**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**HÀ ĐĂNG LƯƠNG**

**1951010032 – DH19CS91**

**TRẦN MINH ĐẠO**

**1951012015 – DHCS91**

**WEBSITE CỬA HÀNG BÁN MÁY TÍNH**

**DYNO STORE**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2023BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**HÀ ĐĂNG LƯƠNG**

**TRẦN MINH ĐẠO**

**WEBSITE CỬA HÀNG BÁN MÁY TÍNH**

**DYNO STORE**

**Mã số sinh viên: 1951010032**

**Mã số sinh viên: 1951012015**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: ThS. HỒ QUANG KHẢI**

**TP. HỒ CHÍ MINH, 2023**

**LỜI CẢM ƠN**

Em xin trân trọng gửi lời cảm ơn đến thầy Hồ Quang Khải, một người luôn luôn sát cánh với các bạn trong việc hướng dẫn cũng như giải đáp các thắc mắc và cho những lời khuyên trong việc xây dựng đề tài giúp xử lý hiệu quả hơn.

Nhờ có sự hướng dẫn, chỉ bảo của thầy/cô mà em đã hoàn thành đồ án một cách thuận lợi và đạt được kết quả khá tốt. Em rất cảm kích những lời khuyên và chỉ dẫn mà thầy đã trao cho em, giúp em tiếp thu được nhiều kiến thức và kinh nghiệm quý giá.

Trong quá trình nhóm em thực hiện đồ án sẽ còn một chút khuyết điểm về mặt ngôn ngữ cũng như cách xử lý chương trình chưa được hoàn thiện. Nên cũng chính vì thế, chúng em mong nhận được sự góp ý từ thầy cô để đề tài của chúng em được hoàn chỉnh và khắc phục các khuyết điểm trong việc xây dựng chương trình cũng như học các kinh nghiệm để hoàn thiện bản thân một cách tốt hơn.

Một lần nữa, em xin chân thành cảm ơn thầy rất nhiều!

Thành phố Hồ Chí Minh, ngày 05 tháng 05 năm 2023

Sinh viên thực hiện

Hà Đăng Lương

Trần Minh Đạo

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

# TÓM TẮT ĐỒ ÁN NGÀNH

1. **Lý do chọn đề tài**

Đề tài “Webste của hàng bán máy tính DYNO Store” với DYNO store là tên của hàng mà tụi em đặt ra và được lựa chọn sau khi đã xem xét và cân nhắc nhiều yếu tố khác nhau.

Trước hết, đây là một đề tài liên quan đến chuyên ngành của tụi em, đó là lập trình web và thiết kế giao diện người dùng. Như vậy, việc nghiên cứu và phát triển một trang web bán hàng đầy đủ chức năng sẽ giúp em củng cố và phát triển kiến thức và kỹ năng của mình trong lĩnh vực này.

Ngoài ra, việc tìm hiểu và áp dụng các phương pháp và kỹ thuật tốt nhất để thiết kế một trang web bán hàng hiệu quả cho cửa hàng máy tính DYNO Store sẽ giúp tăng cường khả năng cạnh tranh của cửa hàng.

Cuối cùng, tụi em muốn nhấn mạnh rằng trong báo cáo đồ án này, tất cả các nội dung và ý tưởng đều là do em tự suy nghĩ, nghiên cứu và viết ra. Chúng em luôn tuân thủ các nguyên tắc đạo đức và trung thực trong học tập và nghiên cứu của mình và không sao chép hay lấy ý tưởng từ bất kỳ nguồn nào mà không được trích dẫn.

1. **Các vấn đề nghiên cứu**

Các công nghệ được dùng để phát triển đồ án bao gồm:

* Django Rest Framework [1]
* ReactJS [2]
* RestFul API [3].

1. **Kết quả**

* Sau khi nghiên cứu cũng như phân tích hiện trạng trong việc xây dựng đề tài

nhằm giải quyết các chức năng cơ bản trong việc quản lý, đồng thời xem và phát triển các chức năng mới cho hệ thống nhằm hoàn thiện hệ thống ứng dụng. Thông qua việc phân tích cũng như lắng nghe sự hướng dẫn của các thầy cô chúng em hoàn thành các mục tiêu đề ra như sau:

* Xác định đối tượng người dùng cũng như việc phân quyền cơ bản cho các đối

tượng riêng biệt như: Admin, Khách hàng.

* Xác định các chức năng cũng như áp dụng các công nghệ trong việc thiết kế ứng dụng.
* Tạo ra các chức năng nhằm đáp ứng nhu cầu cho người dùng dễ dàng sử dụng phần mềm.

1. **Hướng phát triển đề tài**

Qua sự tìm hiểu và nhu cầu thị hiếu trong việc đáp ứng nhu cầu mua hàng giữa

người dùng và hệ thống nhằm giúp hệ thống tăng thêm phương thức mua hàng qua website, giúp người dùng dễ dàng tìm kiếm cũng như thánh toán mặt hàng mình muốn trực tuyến, giảm thiểu sự quá tải trong việc chỉ mua hàng thông qua hình thức offline. Qua đó nhằm thúc đẩy việc phát triển kinh tế nước nhà.

1. **Bố cục phát triển đồ án**

Nội dung báo cáo đồ án chúng em nghiên cứu để phát triển đề tài Hệ thống website bán hàng bao gồm:

* + Chương 1: Tổng quan về đề tài.
  + Chương 2: Giới thiệu khái quát thư viện ReactJS.
  + Chương 3: Giới thiệu khái quát thư viện Django Rest Framework và RestFul Api
  + Chương 4: Phân tích hệ thống
  + Chương 5: Thiết kế hệ thống

**MỤC LỤC**

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 4](#_Toc134222477)

[TÓM TẮT ĐỒ ÁN NGÀNH 5](#_Toc134222478)

[MỤC LỤC 7](#_Toc134222479)

[DANH MỤC TỪ VIẾT TẮT 10](#_Toc134222480)

[DANH MỤC HÌNH VẼ 11](#_Toc134222481)

[DANH MỤC BẢNG 12](#_Toc134222482)

[MỞ ĐẦU 13](#_Toc134222483)

[Chương 1. TỔNG QUAN VỀ ĐỀ TÀI 14](#_Toc134222484)

[1.1. Giới thiệu 14](#_Toc134222485)

[1.2. Mục tiêu của đề tài 14](#_Toc134222486)

[1.3. Phạm vi 14](#_Toc134222487)

[1.4. Các chức năng cơ bản cần có trong chương trình 15](#_Toc134222488)

[1.4.1 Đối với hệ thống dữ liệu 15](#_Toc134222489)

[1.4.2 Đối với người sử dụng 15](#_Toc134222490)

[1.4.3 Đối với người quản lý 15](#_Toc134222491)

[Chương 2: GIỚI THIỆU KHÁI QUÁT THƯ VIỆN REACT JS 16](#_Toc134222492)

[2.1. React là gì ? 16](#_Toc134222493)

[2.2. React JS 16](#_Toc134222494)

[2.2.1 React JS là gì ? 16](#_Toc134222495)

[2.2.2 Ưu và nhược điểm 17](#_Toc134222496)

[2.3 React Native 17](#_Toc134222497)

[2.3.1 React Native là gì ? 17](#_Toc134222498)

[2.3.2 Ưu và nhược điểm 18](#_Toc134222499)

[2.4 Life Cycle Component trong React.JS 18](#_Toc134222500)

[2.4.1 Life Cycle React là gì ? 18](#_Toc134222501)

[2.4.2 Constructor 20](#_Toc134222502)

[2.4.3 DidMount () 20](#_Toc134222503)

[2.4.4 WillUnmount () 21](#_Toc134222504)

[2.4.5 DidUpdate () 22](#_Toc134222505)

[Chương 3: GIỚI THIỆU KHÁI QUÁT THƯ VIỆN DJANGO REST FRAMEWORK VÀ RESTFUL API 23](#_Toc134222506)

[3.1 Django 23](#_Toc134222507)

[3.1.1 Django là gì? 23](#_Toc134222508)

[3.1.2 Ưu điểm và nhược điểm 23](#_Toc134222509)

[3.2 Django Rest Framework 24](#_Toc134222510)

[3.2.1 Django Rest Framework là gì? 24](#_Toc134222511)

[3.2.2 Ưu điểm và nhược điểm 24](#_Toc134222512)

[3.3 RESTful API 25](#_Toc134222513)

[3.3.1 RESTful API là gì? 25](#_Toc134222514)

[3.3.2 Ưu điểm và nhược điểm 25](#_Toc134222515)

[Chương 4: PHÂN TÍCH HỆ THỐNG 26](#_Toc134222516)

[4.1 Các chức năng hệ thống cần có 26](#_Toc134222517)

[4.1.1 Chức năng đặt hàng (hình thức online) 26](#_Toc134222518)

[4.1.2 Chức năng quản lý 26](#_Toc134222519)

[4.1.3 Các dữ liệu mà hệ thống cần lưu 26](#_Toc134222520)

[4.2 Mô tả nghiệp vụ 27](#_Toc134222521)

[4.2.1 Xem sản phẩm 27](#_Toc134222522)

[4.2.2 Đặt hàng 27](#_Toc134222523)

[4.2.3 Thanh toán 27](#_Toc134222524)

[4.2.4 Quản lý giỏ hàng 27](#_Toc134222525)

[4.2.5 Quản lý tài khoản 27](#_Toc134222526)

[4.2.6 Giảm giá và khuyến mãi 28](#_Toc134222527)

[4.2.7 Quản lý đơn hàng 28](#_Toc134222528)

[4.2.8 Giao hàng 28](#_Toc134222529)

[4.2.9 Hỗ trợ khách hàng 28](#_Toc134222530)

[4.3 Vị trí và chức năng xây dựng dự kiến 29](#_Toc134222531)

[Chương 5: Thiết kế hệ thống 30](#_Toc134222532)

[5.1 Thiết kế mô hình BPM 30](#_Toc134222533)

[5.1.1. Đăng nhập 30](#_Toc134222534)

[5.1.2 Đăng ký 31](#_Toc134222535)

[5.1.3 Cập nhật thông tin 31](#_Toc134222536)

[5.1.4 Đặt hàng 32](#_Toc134222537)

[5.1.5 Thanh toán 33](#_Toc134222538)

[5.1.6 Thống kê 34](#_Toc134222539)

[5.2 Thiết kê cơ sở dữ liệu 34](#_Toc134222540)

[5.2.1 Chi tiết các bảng dữ liệu 34](#_Toc134222541)

[5.2.2 Cơ sở dữ liệu quan hệ 40](#_Toc134222542)

[TÀI LIỆU THAM KHẢO 41](#_Toc134222543)

[PHỤ LỤC 42](#_Toc134222544)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Từ đầy đủ |
| React.JS / ReactJs | React Java Script |
| JSX | JavaScript Syntax Extension |
| JS | Java Script |
| DOM | Document Object Model |
| HTML | Hyper Text Markup Language |
| LCR | Life Cycle React |
| props | Properties |
| DRF | Django Rest Framework |
| SQL | Structured Query Language |

DANH MỤC HÌNH VẼ

[*Hình 1: Diagram Component của React.JS* 19](#_Toc134221811)

[*Hình 2: Constructor* 20](#_Toc134221812)

[*Hình 3: WillUnMount ()* 21](#_Toc134221813)

[*Hình 4: DidUpdate* 22](#_Toc134221814)

[*Hình 5: Đăng nhập giữa người dùng và hệ thống* 30](#_Toc134221815)

[*Hình 6: Đăng ký giữa người dùng và hệ thống* 31](#_Toc134221816)

[*Hình 7: Cập nhật thông tin* 31](#_Toc134221817)

[*Hình 8: Đặt hàng* 32](#_Toc134221818)

[*Hình 9: Thanh toán* 33](#_Toc134221819)

[*Hình 10: Thống kê* 34](#_Toc134221820)

[*Hình 11: Cơ sở dữ liệu quan hệ* 40](#_Toc134221821)

DANH MỤC BẢNG

[Bảng 1: Account 35](#_Toc134221822)

[Bảng 2: User 35](#_Toc134221823)

[Bảng 3: Role 36](#_Toc134221824)

[Bảng 4: Category 36](#_Toc134221825)

[Bảng 5: Product 37](#_Toc134221826)

[Bảng 6: Order 38](#_Toc134221827)

[Bảng 7: Order\_product 39](#_Toc134221828)

[Bảng 8: Voucher 39](#_Toc134221829)

MỞ ĐẦU

Trong thời đại công nghệ phát triển như hiện nay, việc sử dụng trang web để thực hiện các giao dịch thương mại đã trở nên phổ biến và tiện lợi hơn bao giờ hết. Đặc biệt, việc mua sắm trực tuyến trở thành xu hướng thịnh hành, giúp cho việc mua bán trở nên dễ dàng hơn, tiết kiệm thời gian và chi phí cho người dùng.

Trong bối cảnh đó, đề tài "Xây dựng website của hàng bán máy tính" được lựa chọn với mong muốn đóng góp phần nào vào việc cung cấp cho người dùng một trải nghiệm mua sắm trực tuyến tốt nhất có thể. Đề tài sẽ tập trung vào xây dựng một trang web bán hàng đầy đủ tính năng, thân thiện và dễ sử dụng cho người dùng, cùng với cơ chế quản lý sản phẩm và đơn hàng hiệu quả cho phía quản trị viên.

Đề tài sẽ tập trung vào xây dựng một trang web bán hàng đầy đủ tính năng, thân thiện và dễ sử dụng cho người dùng, cùng với cơ chế quản lý sản phẩm và đơn hàng hiệu quả cho phía quản trị viên. Chúng em sẽ sử dụng các kỹ thuật và công nghệ mới nhất để thiết kế và phát triển trang web, đảm bảo tính ổn định và an toàn cho người dùng. Đồng thời, em cũng sẽ tập trung vào tối ưu hóa trải nghiệm người dùng, tăng cường tính tương tác và động lực mua sắm, giúp cho người dùng cảm thấy thỏa mãn và hài lòng hơn với trang web của chúng em.

Với mong muốn tạo ra một sản phẩm có chất lượng tốt nhất cho khách hàng, chúng em cam kết sẽ tôn trọng đạo đức học thuật và sẽ không sao chép hay lấy ý tưởng từ bất kỳ nguồn nào mà không được trích dẫn.

# Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu

Đề tài "Xây dựng website bán máy tính" nhằm tạo ra một trang web kinh doanh máy tính. Trang web cung cấp các chức năng chính bao gồm:

* Đăng ký và đăng nhập tài khoản.
* Xem thông tin sản phẩm.
* Sắp xếp và tìm kiếm sản phẩm.
* Thêm sản phẩm vào giỏ hàng và thanh toán.
* Quản lý giỏ hàng và lịch sử đơn hàng.
* Quản lý sản phẩm và đơn hàng từ phía quản trị viên.

Trang web được thiết kế với giao diện thân thiện, dễ sử dụng và tương thích với các thiết bị di động. Công nghệ sử dụng bao gồm:

* Front End: ReactJS.
* Back End: Django Rest Framework, RestfulAPI, MySQL.

Trong quá trình phát triển, chú trọng vào tối ưu hóa hiệu suất và thời gian tải trang, nhằm cung cấp trải nghiệm mua hàng tốt nhất cho người dùng

## Mục tiêu của đề tài

Mục tiêu của đề tài là xây dựng trang web bán máy tính, có các tính năng đơn giản giúp khách hàng dễ dàng tìm kiếm, chọn sản phẩm và thanh toán với thời gian nhanh chóng. Đồng thời, đề tài tập trung xây dựng cơ chế quản lý sản phẩm và đơn hàng cho phía quản trị viên được hiệu quả và dễ quản lý.

## Phạm vi

Phạm vi của đề tài bao gồm các giai đoạn sau:

* Thiết kế và cài đặt giao diện web gồm trang chủ, sản phẩm, giỏ hàng, thanh toán và trang quản trị.
* Xây dựng cơ sở dữ liệu để lưu trữ thông tin sản phẩm, khách hàng, đơn hàng và tài khoản người dùng.
* Cài đặt và triển khai các chức năng cơ bản như sắp xếp, tìm kiếm sản phẩm, đăng ký và đăng nhập tài khoản, thêm vào giỏ hàng, thanh toán, quản lý giỏ hàng và lịch sử đơn hàng.
* Tối ưu hiệu suất của trang web và xây dựng cơ chế quản lý sản phẩm và đơn hàng hợp lý từ phía quản trị viên.

## Các chức năng cơ bản cần có trong chương trình

### Đối với hệ thống dữ liệu

Cần được rõ ràng, độ chính xác cao, thao tác dễ sử dụng.

### Đối với người sử dụng

* Giao diện trực quan, dễ tiếp cận.
* Thuận lợi trong việc tìm kiếm và thanh toán sản phẩm, xem lịch sử đơn hàng.
* Có thể liên lạc với nhân viên khi cần hỗ trợ.
* Dễ dàng thanh toán bằng phương thức online.

### Đối với người quản lý

* Có thể cập nhật, thêm, sửa, xóa các mục quản lý như: khách hàng, danh mục, sản phẩm...
* Quản lý, giám sát hệ thống một cách dễ dàng.
* Thống kê và báo cáo

# Chương 2: GIỚI THIỆU KHÁI QUÁT THƯ VIỆN REACT JS

## 2.1. React là gì ?

React được cho biết là một thư viện JS front – end, một thư viện – mã nguồn mở miễn phí cho lập trình viên sử dụng để xây dựng giao diện người dùng dựa trên các thành phần UI riêng lẻ có sẵn cho lập trình viên front-end.

React được phát triển và duy trì bởi Meta (trước đây là Facebook) cùng các cộng đồng nhà phát tiển và các công ty cá nhân.

React được sử dụng để phát triển các ứng dụng SPA (single-page), các thiết bị di động và ứng dụng kết xuất bằng máy chủ với đa dạng thư viện khác nhau như Next.js và được cài đặt trên môi trường Node.js.

Với việc tìm hiểu và qua các thông tin trên mạng, React có 2 nền tảng, đó là:

* React JS: dành cho nền tảng web.
* React Native: dành cho phần mềm trên nền tảng thiết bị di động.

## 2.2. React JS

### React JS là gì ?

Có thể biết React (còn được gọi là Reactjs hay React.js) một thư viện javascript cho phép những nhà phát triển tạo ra một trang web với giao diện người dùng nhanh chóng.

Đa phần React sử dụng chủ yếu là các component mà cụ thể là các component này chứa các thẻ HTML. Điểm thú vị và đặc trưng của Reactjs là việc render dữ liệu không chỉ ở server mà còn ở cả client.

Ngoài ra React còn sử dụng một khái niệm khác nữa đó là Virtual DOM (DOM ảo). Vậy nó có tác dụng gì ? Virtual DOM tạo ra bản cache cấu trúc dữ liệu trong bộ nhớ, và cứ mỗi vòng lặp nó sẽ liệt kê sự thay đổi và cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả nhất. Mặc dù thực tế chỉ render lại những component hay các subcomponent nào có sự thay đổi.

### Ưu và nhược điểm

|  |  |
| --- | --- |
| ***Ưu điểm*** | ***Nhược điểm*** |
| Cải thiện hiệu suất nhờ DOM | Reactjs chỉ phục vụ cho tầng View |
| Code các đoạn JS dễ dàng hơn | Khi tích hợp Reactjs vào các framework cần phải cấu hình lại |
| Sử dụng cú pháp đặc biệt là JSX (trộn giữa HTML và JS) | Nặng hơn so với các framework khác |
| Render tầng server | Khó tiếp cận cho người mới học về Web |
| Dễ dàng viết các test case giao diện |  |
| Hiệu năng cao cho các ứng dụng có dữ liệu liên tục thay đổi |  |
| Rất dễ cho việc bảo trì và sữa lỗi |  |

## React Native

### React Native là gì ?

Là công cụ giúp lập trình viên lập trình đa nền tảng để tạo ra các ứng dụng trên môi trường native.

Là một framework mã nguồn mở, được viết chủ yếu dựa trên javascript tạo ra các phần mềm trên điện thoại di động Android và IOS.

React Native có thể nói là khá giống React, thay vì sử dụng các web components thì sử dụng các native components

Và để có thể code được cũng như là hiểu được cấu trúc của React native thì chúng ta cần phải nắm được các kiến thức căn bản về JavaScript, những khái niệm cở bản của React như JSX, các state, props và các components, …

### Ưu và nhược điểm

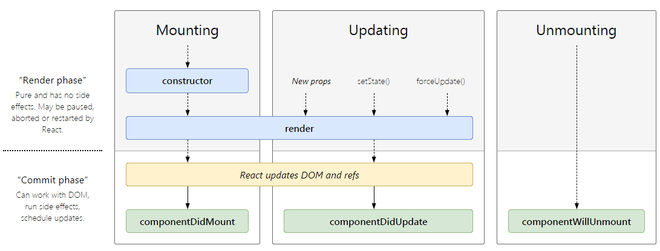
|  |  |
| --- | --- |
| ***Ưu điểm*** | ***Nhược điểm*** |
| Khả năng tái sử dụng code và các components | Hiệu năng kém hơn với Native App |
| Có cộng đồng dev mạnh mẽ | Do sử dụng ngôn ngữ JS nên bảo mật người dùng chưa thật sự tốt |
| Ổn định và tối ưu, trải nghiệm người dùng tốt hơn | Quản lí bộ nhớ |
|  | Không thích hợp cho các app dùng để tính toán cao (crypto, has, ...) |

## Life Cycle Component trong React.JS

### Life Cycle React là gì ?

Life Cycle React có thể được hiểu là cách hoạt động của các component bên trong ReactJS. Gồm có 3 giai đoạn:

* **Mounting ( Tạo ra )**
* **Updating ( Thay đổi )**
* **UnMounting ( Hủy Bỏ )**



*Hình 1: Diagram Component của React.JS*

Qua hình trên ta có thể thấy được một life cycle của component được chạy như thế nào:

* **Mouting** gọi => constructor, render và DidMount
* **Updating** gọi => render và DidUpdate
* **Unmounting** gọi **=>** WillUnmount

Dưới đây là chi tiết các hàm sẽ được sử dụng khi code một web có sử dụng thư viện React

### Constructor

|  |  |
| --- | --- |
| *Hình 2: Constructor* | - Khi một component được khởi tạo, thì lúc này component mounting sẽ được đi qua hàm gọi là constructor  - Trong constructor này sẽ chứa các khai báo (state); các props (thuộc tính) dùng để khởi tạo của component. |

### DidMount ()

Khi một component được Mounting (tạo ra) sẽ được đi qua hàm Constructor để render lần đầu tiên, để hiển thị lên giao diện của người dùng.

Thông thường hàm DidMount () sẽ đảm nhận vai trò gọi API để lấy dữ liệu, sau đó sẽ được binding (ràng buộc) thông qua các setState để cập nhật state

Ví dụ thông thường ở React ES6 version >với cú pháp setState được khai báo như sau:

* const [state, setState] = useState (initialStateValue);

Trong đó:

* state: là giá trị sẽ được binding hay còn gọi là tham số của useState
* setState: là định nghĩa cho tên của một fuction và dùng cho việc update state và là tham số của useState
* initialStateValue: là giá trị ban đầu của state; có thể là Object, mảng, chuỗi, Boolean, number,… tùy vào cách người sử dụng khai báo và những giá trị này sẽ là giá trị của state.
* useState là một hook cơ bản của reactjs cho phép chúng ta khai báo local state trong Function Component và với phiên bản mới nhất này khi sử dụng hook chúng ta sẽ nhanh gọn hơn với những phiên bản cũ khi phải dùng Class Component

### WillUnmount ()

|  |  |
| --- | --- |
| *Hình 3: WillUnMount ()* | Đây là khi một unmounting component được hủy thì hàm willUnmount () sẽ được gọi.  Điều này xảy ra khi ta không render component hoặc chuyển trang thì sẽ được hủy bỏ và bắt đầu một vòng life cycle mới để tải nội dung mới lên  Để hủy việc render này chúng ta có thể sử dụng các hàm được tích hợp có sẵn trong thư viện ReactJS như setTimeout(); setInterval() => tránh trường hợp bị render quá nhiều lần dẫn đến tình trạng bị leak memory (rò rỉ bộ nhớ) khiến cho web bị treo hoặc văng ra. |

**Lưu ý:** hàm WillUnmount() chỉ có thể chạy được một lần duy nhất khi đang trong vòng đời của react component. Tương tự như vậy Mount() cũng chỉ có thể chạy 1 lần duy nhất. Còn riêng DidUpdate() có thể gọi nhiều lần nếu có nhiều update cho component

### DidUpdate ()

|  |  |
| --- | --- |
| *Hình 4: DidUpdate* | Sử dụng khi có sự thay đổi, lúc này component sẽ gọi hàm DidUpdate ()  Có thể hàm này sẽ không được gọi, cũng có thể gọi nhiều lần khi có sự thay đổi, update của component  DidUpdate() sẽ cập nhật khi các props, state thay đổi hoặc update bắt buộc (forceUpdate) |

**Lưu ý:** khi sử dụng DidUpdate () cần phải quản lý chặt chẽ các sự thay đổi của props, state. Vì nếu như không quản lí chặt chẽ sẽ dẫn đến việc render quá nhiều lần gây chậm, hoặc có thể gây ra tình trạng lặp vô hạn do sự thay đổi trong state của hàm DidUpdate ()

# Chương 3: GIỚI THIỆU KHÁI QUÁT THƯ VIỆN DJANGO REST FRAMEWORK VÀ RESTFUL API

## Django

### Django là gì?

Django là một framework web được phát triển bằng ngôn ngữ lập trình Python, giúp cho việc xây dựng các ứng dụng web dễ dàng, nhanh chóng và linh hoạt hơn. Nó cung cấp các tính năng quan trọng như quản lý cơ sở dữ liệu, xử lý yêu cầu HTTP, xử lý mã HTML/CSS/JS và an toàn bảo mật. Với khả năng phát triển nhanh, Django giúp người phát triển có thể làm việc hiệu quả, xây dựng sản phẩm nhanh chóng và dễ dàng triển khai ứng dụng. Django là một framework phổ biến và đã được sử dụng trong nhiều dự án lớn trên toàn thế giới.

### Ưu điểm và nhược điểm

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| Tính năng cần thiết | Cung cấp đầy đủ các tính năng cần thiết để xây dựng ứng dụng web | Không hỗ trợ các tính năng nâng cao như NodeJS hoặc Ruby on Rails |
| Cấu trúc dự án mẫu | Có cấu trúc dự án mẫu đucợ chuẩn hóa, thân thiện với người dùng, giúp cho người dùng dễ dàng quản lý và phát triển dự án | Khó phù hợp với các dự án nhỏ |
| Bảo mật | Được tích hợp các tính năng bảo mật mạnh mẽ như xác thực, phân quyền truy cập, giúp cho ứng dụng của bạn an toàn hơn | Khó sử dụng trong các ứng dụng có tính bảo mật cao |
| Tích hợp dễ dàng | Django tích hợp dễ dàng với các công nghệ như ReactJS,Angular hoặc VueJS | Khó tùy biến khi phải xử lý các tình huống đặc biệt |

## Django Rest Framework

### Django Rest Framework là gì?

Django Rest Framework (DRF) là một thư viện Python dùng để xây dựng các API RESTful trong Django. Thư viện này cung cấp các công cụ và thư viện giúp lập trình viên xử lý dữ liệu, xác thực người dùng, phân trang và thực hiện các thao tác CRUD trên cơ sở dữ liệu. Nó đang được sử dụng phổ biến trong việc xây dựng các ứng dụng web và di động, đặc biệt là ứng dụng truyền thông xã hội đa phương tiện, ứng dụng IoT và các ứng dụng yêu cầu tích hợp giữa các nền tảng khác nhau.

### Ưu điểm và nhược điểm

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| Tính năng cần thiết | Cung cấp đầy đủ các tính năng cần thiết để phát triển các ứng dụng RESTfull API | Khó sử dụng với các ứng dụng không phải là RESTfull API |
| Hỗ trợ xác thực | Cung cấp tốt các tính năng xác thực tốt, bao gồm cả JWT | Cấu hình xác thực khá phức tạp và khó sử dụng |
| Hiệu suất | Có hiệu suất cao do sử dụng serializer để tối ưu việc tải dữ liệu | Yêu cầu cho phép thông tin trải dài trên nhiều trang (pagination) khá cao |
| Tích hợp dễ dàng | Django tích hợp dễ dàng với các công nghệ bên ngoài như ReactJS, Angular hoặc VueJS | Khó tùy biến khi phải xử lý các tình huống đặc biệt |

## RESTful API

### RESTful API là gì?

RESTful API (Representational State Transfer) là một kiểu thiết kế các ứng dụng web theo một tập hợp các quy tắc mà khi tuân thủ các quy tắc đó, thì ứng dụng có thể hỗ trợ điều hướng dữ liệu và hoạt động một cách hiệu quả.

### Ưu điểm và nhược điểm

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Dễ hiểu | Khó phát hiện và maintain |
| Có thể cache | Không hỗ trợ transaction |
| Có khả năng mở rộng | Khó sử dụng cho các ứng dụng real-time |
| Sử dụng các phương thức HTTP (GET, POST, PUT, DELETE) | Khó thực hiện các hoạt động cộng tác |
| Hỗ trợ công nghệ web mới nhất |  |
| Chia sẻ tài nguyên |  |

# Chương 4: PHÂN TÍCH HỆ THỐNG

## Các chức năng hệ thống cần có

### Chức năng đặt hàng (hình thức online)

* Đối tượng: khách hàng.
* Chức năng:
* Đăng nhập/đăng ký tài khoản.
* Xem thông tin tài khoản khách hàng.
* Tìm/ xem thông tin sản phẩm theo tên, giá.
* Đặt hàng.
* Thanh toán.
* Hủy đơn hàng.

### Chức năng quản lý

* Đối tượng: quản trị (admin)
* Chức năng:
* Đăng nhập theo tài khoản quản trị.
* Quản lý hệ thống:
* Quản lý tài khoản
* Quản lý sản phẩm
* Quản lý đơn hàng
* Quản lý danh mục

### Các dữ liệu mà hệ thống cần lưu

* Thông tin khách hàng
* Thông tin danh mục
* Thông tin sản phẩm
* Thông tin đơn hàng
* Thông tin thanh toán
* Thông tin đánh giá

## Mô tả nghiệp vụ

### Xem sản phẩm

* Khi truy cập vào website bán hàng, khách hàng có thể xem các sản phẩm theo danh sách hoặc tìm kiếm sản phẩm cụ thể bằng cách nhập từ khóa vào thanh tìm kiếm. Chức năng phân loại sản phẩm cũng giúp khách hàng dễ dàng tìm kiếm các sản phẩm theo danh mục cụ thể. Các thông tin chi tiết của sản phẩm gồm hình ảnh, mô tả, giá cả, thông số kỹ thuật, và các chương trình khuyến mãi.

### Đặt hàng

* Khi khách hàng chọn sản phẩm cần mua, họ có thể thêm vào giỏ hàng và tiếp tục chọn sản phẩm khác nếu muốn. Sau khi hoàn tất việc đặt hàng, khách hàng sẽ nhập các thông tin mua hàng như số lượng sản phẩm, địa chỉ giao hàng và phương thức thanh toán. Các thông tin này được lưu trữ và xử lý để sản phẩm có thể được giao đến địa chỉ xác định.

### Thanh toán

* Khách hàng có thể chọn thanh toán qua nhiều phương thức khác nhau như thẻ tín dụng, ví điện tử hoặc COD (thanh toán tiền mặt khi nhận hàng). Website bán hàng cung cấp các cổng thanh toán để khách hàng dễ dàng thanh toán chính xác.

### Quản lý giỏ hàng

* Trong quá trình mua hàng, khách hàng có thể xem lại thông tin giỏ hàng và thêm hoặc xóa sản phẩm trước khi thanh toán. Thông tin về giỏ hàng của khách hàng sẽ được lưu trữ để họ có thể quay lại mua hàng sau này.

### Quản lý tài khoản

* Khách hàng có thể tạo tài khoản để quản lý thông tin cá nhân, địa chỉ giao hàng, lịch sử đặt hàng và các thông tin khác. Việc đăng ký tài khoản giúp khách hàng thuận tiện hơn trong quá trình mua hàng và giúp cho việc quản lý thông tin hàng hóa nhanh chóng và chính xác hơn.

### Giảm giá và khuyến mãi

* Website bán hàng cung cấp các chương trình giảm giá và khuyến mãi để thu hút khách hàng và tăng doanh số bán hàng. Điều này giúp khách hàng có thể mua các sản phẩm với giá tốt hơn và cùng lúc tăng thu nhập cho nhà bán hàng.

### Quản lý đơn hàng

* Người quản lý trang web có thể kiểm tra đơn hàng, lưu trữ thông tin đơn hàng và cập nhật trạng thái đơn hàng. Thông tin đơn hàng sẽ được lưu trữ và sắp xếp để tiện quản lý và theo dõi quá trình giao hàng.

### Giao hàng

* Sau khi đơn hàng được xử lý và thanh toán, sản phẩm được đóng gói và giao đến địa chỉ khách hàng. Thông tin về đơn hàng và trạng thái giao hàng được cập nhật để khách hàng có thể kiểm tra và theo dõi quá trình giao hàng.

### Hỗ trợ khách hàng

* Website bán hàng cung cấp các kênh liên lạc để giúp khách hàng giải đáp các thắc mắc hoặc giải quyết các vấn đề phát sinh. Các kênh liên lạc bao gồm email, điện thoại. Một số website bán hàng còn có các trang hướng dẫn sử dụng sản phẩm, câu hỏi thường gặp và các thông tin hữu ích khác để khách hàng có thể tìm kiếm thông tin mà họ cần một cách nhanh chóng và dễ dàng.

## Vị trí và chức năng xây dựng dự kiến

|  |  |
| --- | --- |
| Vị trí | Chức năng |
| Người dùng (chưa có tài khoản) | - Xem trang chủ  - Xem giới thiệu công ty  - Đăng ký tài khoản khách hàng |
| Người dùng (đã có tài khoản) | - Xem trang chủ  - Xem giới thiệu công ty  - Đăng nhập  - Đăng ký tài khoản khách hàng  - Xem thông tin tài khoản  - Chỉnh sửa tài khoản  - Tìm kiếm sản phẩm  - Thanh toán online hoặc offline  - Xem lịch sử đặt hàng |
| Admin | - Xem trang chủ Admin  - Quản lý thêm, sửa, xóa danh mục, sản phẩm và người dùng  - Thống kê đơn hàng |

# Chương 5: Thiết kế hệ thống

## Thiết kế mô hình BPM

### 5.1.1. Đăng nhập

A picture containing text, screenshot, diagram, font

Description automatically generated

*Hình 5: Đăng nhập giữa người dùng và hệ thống*

### 5.1.2 Đăng ký

A screenshot of a diagram

Description automatically generated with low confidence

*Hình 6: Đăng ký giữa người dùng và hệ thống*

### 5.1.3 Cập nhật thông tin

A picture containing text, diagram, line, font

Description automatically generated

*Hình 7: Cập nhật thông tin*

### 5.1.4 Đặt hàng

A screenshot of a computer

Description automatically generated with low confidence

*Hình 8: Đặt hàng*

### 5.1.5 Thanh toán

A picture containing text, screenshot, diagram, software

Description automatically generated

*Hình 9: Thanh toán*

### Thống kê

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 10: Thống kê*

## 5.2 Thiết kê cơ sở dữ liệu

### Chi tiết các bảng dữ liệu

#### 5.2.1.1 Bảng account

Bảng này được thiết kế lấy tất cả các thông tin của người dùng. Với thuộc tính role\_id cho phép người dùng sử dụng các chức năng riêng biệt theo từng role đã

được phân quyền khi tham gia sử dụng vào hệ thống.

Bảng : Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | Id | INT | Mã account | primary key |
| 2 | Username | VARCHAR(255) | Tên đăng nhập |  |
| 3 | Password | VARCHAR(255) | Mật khẩu |  |
| 4 | Role\_id | CHAR(32) | Quyền sử dụng |  |
| 5 | Is\_active | INT(1) | Tình trạng |  |
| 6 | Created\_at | DATETIME(6) | Ngày tạo |  |
| 7 | Update\_at | DATETIME(6) | Ngày cập nhật |  |

#### 5.2.1.2 Bảng user

Bảng này sẽ lưu tất cả thông tin người dùng dưới mã account\_id.

Bảng : User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | Name | VARCHAR(100) | Tên người dùng |  |
| 2 | Phone | VARCHAR(20) | Số điện thoại |  |
| 3 | Birthday | DATE | Ngày sinh |  |
| 4 | Email | VARCHAR(100) | Email |  |
| 5 | Gender | VARCHAR(10) | Giới tính |  |
| 6 | Account\_id | INT | Mã account | Foreign key |

#### Bảng role

Bảng này sẽ lưu dữ liệu dưới vai trò của người dùng.

Bảng : Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | Id | INT | Mã phân quyền | primary key |
| 2 | Name | VARCHAR(50) | Quyền |  |
| 3 | created\_at | DATETIME(6) | Ngày tạo |  |
| 4 | updated\_at | DATETIME(6) | Ngày cập nhật |  |
| 5 | is\_activate | INT(1) | Tình trạng |  |

#### Bảng category

Bảng này sẽ lưu tất cả thông tin danh mục

Bảng : Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | id | INT | Mã danh mục | primary key |
| 2 | Name | VARCHAR(255) | Tên danh mục |  |
| 3 | slug | VARCHAR(50) | Mô tả |  |
| 4 | created\_at | DATETIME(6) | Ngày tạo |  |
| 5 | updated\_at | DATETIME(6) | Ngày cập nhật |  |
| 6 | is\_activate | INT(1) | Tình trạng |  |

#### Bảng product

Bảng lưu tất cả thông tin sản phẩm với trường category\_id cho biết sản phẩm thuộc danh mục nào.

Bảng : Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | Id | INT | Mã sản phẩm | primary key |
| 2 | Name | VARCHAR(255) | Tên sản phẩm |  |
| 3 | Slug | VARCHAR(255) | Mô tả |  |
| 4 | description | LONGTEXT | Mô tả chi tiết sản phẩm |  |
| 5 | price | DECIMAL(6,2) | Giá sản phẩm |  |
| 6 | image | VARCHAR(100) | ảnh sản phẩm |  |
| 7 | thumbnail | VARCHAR(100) | ảnh nhỏ |  |
| 8 | category\_id | INT | Mã danh mục | Foreign key |
| 9 | created\_at | DATETIME | Ngày tạo |  |
| 10 | updated | DATETIME | Ngày cập nhật |  |
| 11 | is\_activate | INT(1) | Tình trạng |  |

#### Bảng order

Bảng này sẽ lưu dữ liệu mã đơn hàng và thông tin/ địa chỉ giao hàng với trường account\_id cho biết đơn hàng này thuộc người dùng nào

Bảng : Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | id | INT | Mã đơn hàng | Primary key |
| 2 | first\_name | VARCHAR(100) | Tên người nhận hàng |  |
| 3 | last\_name | VARCHAR(100) | Họ |  |
| 4 | email | VARCHAR(100) | Email |  |
| 5 | address | VARCHAR(100) | Địa chỉ giao hàng |  |
| 6 | zipcode | VARCHAR(100) | Mã bưu điện |  |
| 7 | place | VARCHAR(100) | Thành phố |  |
| 8 | phone | VARCHAR(100) | Số điện thoại nhận hàng |  |
| 9 | paid\_amount | DECIMAL(8,2) | Số tiền phải trả |  |
| 10 | stripe\_token | VARCHAR(100) | Mã token |  |
| 11 | account\_id | INT | Mã người dùng | Foreign key |
| 12 | created\_at | DATETIME(6) | Ngày tạo |  |
| 13 | updated\_at | DATETIME(6) | Ngày cập nhật |  |
| 14 | is\_activate | INT(1) | Tình trạng |  |

#### Bảng order\_product

Bảng này sẽ lưu thông tin chi tiết đơn hàng

Bảng : Order\_product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | Id | INT | Mã thông tin đơn hàng | Primary key |
| 2 | Price | DECIMAL(8,2) | Giá sản phẩm |  |
| 3 | quantity | INT | Số lượng |  |
| 4 | order\_id | INT | Mã đơn hàng | Foreign key |
| 5 | created\_at | DATETIME(6) | Ngày tạo |  |
| 6 | updated\_at | DATETIME(6) | Ngày cập nhật |  |
| 7 | is\_activate | INT(1) | Tình trạng |  |

#### Bảng voucher

Bảng này sẽ lưu voucher và số tiền được giảm giá cho mỗi voucher

Bảng : Voucher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả | Ràng buộc |
| 1 | Id | INT | Mã voucher | Primary key |
| 2 | Name | VARCHAR(100) | Tên voucher |  |
| 3 | Rate | DECIMAL(4,2) | Tỉ lệ giảm giá |  |
| 4 | created\_at | DATETIME(6) | Ngày tạo |  |
| 5 | updated\_at | DATETIME(6) | Ngày cập nhật |  |
| 6 | is\_activate | INT(1) | Tình trạng |  |

### Cơ sở dữ liệu quan hệ

A picture containing text, screenshot, number, font

Description automatically generated

*Hình 11: Cơ sở dữ liệu quan hệ*

TÀI LIỆU THAM KHẢO

[ReactJS là gì? Những điều có thể bạn chưa biết về ReactJS (200lab.io)](https://200lab.io/blog/reactjs-la-gi/)

[React Native là gì? Có nên học react native? Tổng quan về React Native (nordiccoder.com)](https://nordiccoder.com/blog/tong-quan-ve-react-native/)

PHỤ LỤC