TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP. HCM

Khoa Công Nghệ Thông Tin

**ĐỀ CƯƠNG ĐỀ ÁN TỐT NGHIỆP**

****

**ĐỀ TÀI:**

**Xây dựng Website bán quần áo (sử dụng Reactjs,Boostrap,java spring boot)**

GVHD: Huỳnh Tấn Phát

SVTH: Trương Hoàng Lâm

MSSV: 2121110229

TPHCM, ngày 24 tháng 01 năm 2024

# LỜI NHẬN XÉT

# LỜI CẢM ƠN

# MỤC LỤC

[LỜI NHẬN XÉT i](#_Toc100257230)

[LỜI CẢM ƠN i](#_Toc100257231)

[MỤC LỤC ii](#_Toc100257232)

[DANH MỤC CÁC BẢNG v](#_Toc100257233)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ v](#_Toc100257234)

[LỜI MỞ ĐẦU v](#_Toc100257235)

[Chương 1. TỔNG QUAN 1](#_Toc100257236)

[1.1. Mern Stack. 1](#_Toc100257237)

[**1.1.1.** **Giới thiệu Reactjs** 1](#_Toc100257238)

[**1.1.2.** **Các thành phần cơ bản của Reactjs** 1](#_Toc100257239)

[1.2. Restful API. 1](#_Toc100257240)

[**1.2.1.** **Giới thiệu Restful API.** 1](#_Toc100257241)

[**1.2.2.** **Các thành phần cơ bản của Restful, Restful API.** 1](#_Toc100257242)

[1.3. Bootstrap. 1](#_Toc100257243)

[**1.3.1.** **Giới thiệu Bootstrap** 1](#_Toc100257244)

[**1.4.2.** **Các thành phần cơ bản của Nodejs.** 1](#_Toc100257248)

[1.5. Express Framework. 1](#_Toc100257249)

[**1.5.1.** **Giới thiệu Express Framework.** 1](#_Toc100257250)

[**1.5.2.** **Cấu trúc của Express Framework.** 1](#_Toc100257251)

[Chương 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 3](#_Toc100257256)

[2.1. Khảo sát và phân tích hệ thống. 3](#_Toc100257257)

[**2.1.1.** **Khảo sát yêu cầu.** 3](#_Toc100257258)

[**2.1.2.** **Giới thiệu hệ thống.** 3](#_Toc100257259)

[**2.1.3.** **Mục tiêu xây dựng hệ thống.** 3](#_Toc100257260)

[**2.1.4.** **Các vai trò của người dùng trong hệ thống.** 3](#_Toc100257261)

[**2.1.5.** **Các yêu cầu chức năng của hệ thống.** 3](#_Toc100257262)

[**2.1.6.** **Các yêu cầu phi chức năng của hệ thống.** 3](#_Toc100257263)

[2.2. Đặc tả và thiết kế hệ thống. 3](#_Toc100257264)

[**2.2.1.** **Đặc tả hệ thống.** 3](#_Toc100257265)

[**2.2.2.** **Biểu đồ use case tổng quát.** 3](#_Toc100257266)

[**2.2.3.** **Biểu đồ use case sử dụng cho nhân viên quản trị.** 3](#_Toc100257267)

[**2.2.4.** **Biểu đồ use case sử dụng cho người dùng.** 3](#_Toc100257268)

[2.3. Biểu đồ lớp. 3](#_Toc100257269)

[2.4. Biểu đồ hoạt động 3](#_Toc100257270)

[**2.4.1.** **Phía khách hàng.** 3](#_Toc100257271)

[**2.4.2.** **Phía quản trị.** 4](#_Toc100257272)

[2.5. Biểu đồ tuần tự. 5](#_Toc100257273)

[**2.5.1.** **Phía khách hàng.** 5](#_Toc100257274)

[**2.5.2.** **Phía quản trị.** 5](#_Toc100257275)

[2.6. Cơ sở dữ liệu. 6](#_Toc100257276)

[**2.6.1.** **Lược đồ liên kết giữa các bảng.** 6](#_Toc100257277)

[**2.6.2.** **Các bảng cơ sở dữ liệu.** 6](#_Toc100257278)

[Chương 3. THIẾT KẾ HỆ THỐNG 8](#_Toc100257279)

[3.1. Triển khai cài đặt. 9](#_Toc100257280)

[**3.1.1.** **Giao diện người dùng*.*** 9](#_Toc100257281)

[**3.1.1.1.** Đăng ký. 9](#_Toc100257282)

[**3.1.1.2.** Đăng nhập. 9](#_Toc100257283)

[**3.1.1.3.** Đăng xuất. 9](#_Toc100257284)

[**3.1.1.4.** Tìm kiếm sản phẩm. 9](#_Toc100257285)

[**3.1.1.5.** Xem thông tin sản phẩm. 9](#_Toc100257286)

[**3.1.1.6.** Xem sản phẩm theo danh mục. 9](#_Toc100257287)

[**3.1.1.7.** Thêm sản phẩm vào giỏ hàng. 9](#_Toc100257288)

[**3.1.1.8.** Cập nhật giỏ hàng. 9](#_Toc100257289)

[**3.1.1.15.** Liên hệ. 9](#_Toc100257296)

[**3.1.1.16.** Xem lịch sử mua hàng. 9](#_Toc100257297)

[**3.1.2.** **Giao diện quản trị*.*** 9](#_Toc100257298)

[**3.1.2.1.** Đăng ký. 9](#_Toc100257299)

[**3.1.2.2.** Đăng nhập. 9](#_Toc100257300)

[**3.1.2.3.** Xem thông tin sản phẩm. 9](#_Toc100257301)

[**3.1.2.4.** Thêm mới sản phẩm 9](#_Toc100257302)

[**3.1.2.5.** Cập nhật sản phẩm. 9](#_Toc100257303)

[**3.1.2.6.** Xóa sản phẩm. 9](#_Toc100257304)

[**3.1.2.7.** Xem thông tin danh mục sản phẩm. 10](#_Toc100257305)

[**3.1.2.8.** Thêm danh mục sản phẩm. 10](#_Toc100257306)

[**3.1.2.9.** Sửa danh mục sản phẩm. 10](#_Toc100257307)

[**3.1.2.10.** Xóa danh mục. 10](#_Toc100257308)

[**3.1.2.11.** Xem thông tin bai viết. 10](#_Toc100257309)

[**3.1.2.12.** Thêm bài viết. 10](#_Toc100257310)

[**3.1.2.13.** Cập nhật bài viết. 10](#_Toc100257311)

[**3.1.2.14.** Xóa bài viết. 10](#_Toc100257312)

[**3.1.2.15.** Xem thông tin đơn hàng. 10](#_Toc100257313)

[**3.1.2.16.** Cập nhật đơn hàng. 10](#_Toc100257314)

[3.2. Đánh giá kết quả. 10](#_Toc100257315)

[**3.2.1.** **Kết quả và hướng phát triển.** 10](#_Toc100257316)

[**3.2.2.** **Hạn chế của đề tài.** 10](#_Toc100257317)

[**KẾT LUẬN** 11](#_Toc100257318)

[TÀI LIỆU THAM KHẢO 12](#_Toc100257319)

# DANH MỤC CÁC BẢNG

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

# LỜI MỞ ĐẦU

# TỔNG QUAN

## Mern Stack.

### **Giới thiệu Reactjs**

ReactJS là một thư viện JavaScript được Facebook phát triển, chủ yếu dùng để xây dựng giao diện người dùng (UI) cho ứng dụng web. Nó sử dụng kiến trúc dựa trên thành phần, có Virtual DOM để tối ưu hiệu suất, và JSX để viết mã nguồn UI dễ đọc. React thực hiện ràng buộc dữ liệu một chiều và luồng dữ liệu một chiều, giúp quản lý trạng thái ứng dụng một cách dễ dàng. Với cộng đồng lớn và hệ sinh thái đa dạng, React đã trở thành công nghệ phổ biến cho phát triển giao diện người dùng web

### **Các thành phần cơ bản của Reactjs**

1. **Component (Thành phần):** Thành phần là đơn vị cơ bản xây dựng giao diện trong React. Mỗi thành phần có thể chứa mã HTML, logic xử lý sự kiện, và có thể được sử dụng tái sử dụng trong ứng dụng.
2. **State (Trạng thái):** State là dữ liệu mà một thành phần giữ để theo dõi trạng thái của nó. Khi state thay đổi, React tự động cập nhật giao diện để phản ánh thay đổi đó.
3. **Props (Thuộc tính):** Props là dữ liệu được truyền từ thành phần cha đến thành phần con. Props giúp truyền dữ liệu giữa các thành phần và làm cho ứng dụng linh hoạt và tái sử dụng được.
4. **Event Handling (Xử lý sự kiện):** React sử dụng cú pháp giống với HTML để xử lý sự kiện, như onClick, onChange, v.v. Xử lý sự kiện giúp đáp ứng khi người dùng tương tác với giao diện.
5. **Lifecycle Methods (Phương thức vòng đời):** React cung cấp các phương thức vòng đời như componentDidMount, componentDidUpdate, componentWillUnmount, v.v. để thực hiện các hành động tại các giai đoạn cụ thể trong quá trình tồn tại của một thành phần.
6. **JSX (JavaScript XML):** JSX là cú pháp mở rộng của JavaScript, cho phép viết mã nguồn UI giống HTML trong mã JavaScript. JSX giúp làm cho mã nguồn dễ đọc và hiểu.
7. **Virtual DOM (DOM ảo):** Virtual DOM là một bản sao của DOM thực tế, giúp React tối ưu hiệu suất bằng cách cập nhật chỉ những phần thay đổi thực sự trên trang

## Restful API.

### **Giới thiệu Restful API.**

RESTful API (Representational State Transfer) là một kiểu kiến trúc thiết kế cho việc xây dựng các dịch vụ web. Nó đặt ra một số nguyên tắc và quy tắc để tạo ra các API (Application Programming Interface) hiệu quả và dễ sử dụng

### **Các thành phần cơ bản của Restful, Restful API.**

1. **Resource (Tài nguyên):** Tài nguyên là đối tượng hoặc dịch vụ mà API cung cấp. Mỗi tài nguyên có một URI (Uniform Resource Identifier) duy nhất để định danh nó. Ví dụ, một tài nguyên có thể là **/users** hoặc **/products**.
2. **URI (Uniform Resource Identifier):** Là định danh duy nhất của mỗi tài nguyên trong hệ thống. URI được sử dụng để truy cập và thao tác trên tài nguyên đó. Ví dụ, **/users/123** có thể là URI của một người dùng có ID là 123.
3. **HTTP Methods (Phương thức HTTP):** RESTful API sử dụng các phương thức HTTP để thực hiện các thao tác trên tài nguyên. Các phương thức phổ biến bao gồm:
   * **GET:** Lấy thông tin về một hoặc nhiều tài nguyên.
   * **POST:** Tạo mới một tài nguyên.
   * **PUT:** Cập nhật thông tin của một tài nguyên hoặc tạo mới nếu chưa tồn tại.
   * **DELETE:** Xóa một tài nguyên.
4. **Representation (Biểu diễn):** Các tài nguyên được biểu diễn dưới dạng dữ liệu có thể truyền tải, thường là JSON hoặc XML. Dữ liệu biểu diễn này chứa thông tin về tài nguyên và các thao tác có thể thực hiện.
5. **Headers (Tiêu đề):** Các tiêu đề HTTP được sử dụng để truyền tải thông tin bổ sung. Ví dụ, **Content-Type** có thể xác định định dạng của dữ liệu biểu diễn, **Authorization** có thể xác thực người dùng.
6. **Status Codes (Mã trạng thái):** Mã trạng thái HTTP được trả về từ server để chỉ ra kết quả của yêu cầu. Mã trạng thái 2xx thường đại diện cho thành công, 4xx cho lỗi từ phía client, và 5xx cho lỗi từ phía server.
7. **Query Parameters (Tham số truy vấn):** Các tham số được gửi kèm theo URL để lọc, sắp xếp, hoặc giới hạn dữ liệu trả về. Ví dụ, **/users?role=admin** có thể được sử dụng để lấy danh sách người dùng với vai trò là admin.
8. **Request Body (Thân yêu cầu):** Đối với các phương thức như POST và PUT, dữ liệu của yêu cầu được chứa trong thân yêu cầu. Định dạng của dữ liệu này thường được xác định bởi tiêu đề **Content-Type**.
9. **Hypermedia (HATEOAS):** Đây là một khái niệm quan trọng trong RESTful, trong đó các tài nguyên trả về cung cấp các liên kết (hyperlinks) để người tiêu dùng có thể khám phá và tương tác với các tài nguyên liên quan mà không cần biết trước về chúng.

## Express Framework.

### **Giới thiệu Express Framework.**

Express là một framework phổ biến được xây dựng trên nền tảng Node.js để phát triển ứng dụng web và API. Nó giúp đơn giản hóa quá trình xây dựng các ứng dụng web bằng cách cung cấp các tính năng và công cụ hữu ích

### **Cấu trúc của Express Framework.**

Cấu trúc của một ứng dụng Express thường được tổ chức dưới dạng mô hình MVC (Model-View-Controller), giúp tách biệt logic của ứng dụng thành các phần riêng biệt. Dưới đây là một cấu trúc thường gặp trong Express:

1. **Thư mục node\_modules:** Chứa các module và thư viện mà ứng dụng sử dụng, được quản lý bởi npm (Node Package Manager).
2. **Tệp package.json:** Chứa thông tin về ứng dụng, các phụ thuộc (dependencies), scripts để chạy và kiểm thử ứng dụng.
3. **Thư mục public:** Chứa các tài nguyên tĩnh như hình ảnh, file CSS, JavaScript, được phục vụ trực tiếp cho client. Các tài nguyên trong thư mục này có thể được truy cập từ phía client thông qua các đường dẫn tương ứng.
4. **Thư mục views:** Chứa các file templates sử dụng để tạo ra giao diện người dùng. Express hỗ trợ nhiều template engine như EJS, Pug (còn gọi là Jade), Handlebars.
5. **Thư mục routes:** Chứa các module định tuyến (route) của ứng dụng. Các route xác định cách các yêu cầu HTTP nên được xử lý và phản hồi.
6. **Thư mục models:** Nếu ứng dụng sử dụng cơ sở dữ liệu, thư mục này thường chứa các module đại diện cho các mô hình dữ liệu (models).
7. **Thư mục controllers:** Chứa các module chứa logic xử lý yêu cầu, thường là nơi xử lý các yêu cầu từ route và tương tác với mô hình (models).
8. **Tệp app.js hoặc index.js:** Là tệp chính khởi động ứng dụng Express. Nó chứa cấu hình chính, khai báo và cấu hình các middleware, thiết lập route, và khởi động server.

## Hệ quản trị cơ sở dữ liệu MySQL

### **Sơ lược về MySQL.**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mã nguồn mở được sử dụng rộng rãi trên toàn thế giới. Được phát triển, duy trì và hỗ trợ bởi Oracle Corporation, MySQL cung cấp một giải pháp lưu trữ dữ liệu mạnh mẽ, linh hoạt và ổn định

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Khảo sát và phân tích hệ thống.

### **Khảo sát yêu cầu.**

Khảo sát yêu cầu cho một trang web bán quần áo là một quá trình quan trọng để đảm bảo rằng hệ thống sẽ đáp ứng đúng nhu cầu và mong đợi của người dùng và doanh nghiệp

### **Giới thiệu hệ thống.**

Một hệ thống bán quần áo trực tuyến với mục tiêu cung cấp trải nghiệm mua sắm thuận lợi và đa dạng cho người dùng. Hệ thống tập trung vào việc cung cấp các xu hướng thời trang mới nhất, sản phẩm chất lượng cao, và dịch vụ chăm sóc khách hàng tốt.

### **Mục tiêu xây dựng hệ thống.**

Cung cấp một trải nghiệm mua sắm trực tuyến tốt nhất cho người dùng, đồng thời đảm bảo sự thuận tiện, đa dạng và an toàn.

### **Các vai trò của người dùng trong hệ thống.**

### **Các yêu cầu chức năng của hệ thống.**

### **Các yêu cầu phi chức năng của hệ thống.**

## Đặc tả và thiết kế hệ thống.

### **Đặc tả hệ thống.**

### **Biểu đồ use case tổng quát.**

### **Biểu đồ use case sử dụng cho nhân viên quản trị.**

### **Biểu đồ use case sử dụng cho người dùng.**

## Biểu đồ lớp.

## Biểu đồ hoạt động

### **Phía khách hàng.**

#### Đăng ký.

#### Đăng nhập.

#### Đăng xuất.

#### Xem thông tin sản phẩm.

#### Xem sản phẩm theo danh mục.

#### Thêm sản phẩm vào giỏ hàng.

#### Cập nhật giỏ hàng.

#### Xem tin tức thời trang.

#### Liên hệ.

#### Xem lịch sử mua hàng.

### **Phía quản trị.**

#### Đăng ký.

#### Đăng nhập.

#### Xem thông tin sản phẩm.

#### Thêm mới sản phẩm

#### Cập nhật sản phẩm.

#### Xóa sản phẩm.

#### Xem thông tin danh mục sản phẩm.

#### Thêm danh mục sản phẩm.

#### Sửa danh mục sản phẩm.

#### Xóa danh mục.

#### Xem thông tin bai viết.

#### Thêm bài viết.

#### Cập nhật bài viết.

#### Xóa bài viết.

#### Xem thông tin đơn hàng.

#### Cập nhật đơn hàng.

## Biểu đồ tuần tự.

### **Phía khách hàng.**

#### Đăng ký.

#### Đăng nhập.

#### Đăng xuất.

#### Tìm kiếm sản phẩm.

#### Xem thông tin sản phẩm.

#### Xem sản phẩm theo danh mục.

#### Thêm sản phẩm vào giỏ hàng.

#### Cập nhật giỏ hàng.

#### Thanh toán.

#### Xem thông tin cá nhân.

#### Xem tin tức thời trang.

#### Tìm kiếm tin tức theo chủ đề.

#### Đánh giá sản phẩm.

#### Thêm sản phẩm vào mục ưa thích.

#### Liên hệ.

#### Xem lịch sử mua hàng.

### **Phía quản trị.**

#### Đăng ký.

#### Đăng nhập.

#### Xem thông tin sản phẩm.

#### Thêm mới sản phẩm

#### Cập nhật sản phẩm.

#### Xóa sản phẩm.

#### Xem thông tin danh mục sản phẩm.

#### Thêm danh mục sản phẩm.

#### Sửa danh mục sản phẩm.

#### Xóa danh mục.

#### Xem thông tin bai viết.

#### Thêm bài viết.

#### Cập nhật bài viết.

#### Xóa bài viết.

#### Xem thông tin đơn hàng.

#### Cập nhật đơn hàng.

## Cơ sở dữ liệu.

### **Lược đồ liên kết giữa các bảng.**

### **Các bảng cơ sở dữ liệu.**

#### Products.

#### Colors.

#### Sizes.

#### Categories.

#### Contacts.

#### Users.

#### Pages.

#### Images.

#### Orders.

#### Topics.

#### Posts.

# THIẾT KẾ HỆ THỐNG

## Triển khai cài đặt.

### **Giao diện người dùng*.***

### Đăng ký.

### Đăng nhập.

### Đăng xuất.

### Tìm kiếm sản phẩm.

### Xem thông tin sản phẩm.

### Xem sản phẩm theo danh mục.

### Thêm sản phẩm vào giỏ hàng.

### Cập nhật giỏ hàng.

### Thanh toán.

### Xem thông tin cá nhân.

### Xem tin tức thời trang.

### Tìm kiếm tin tức theo chủ đề.

### Đánh giá sản phẩm.

### Thêm sản phẩm vào mục ưa thích.

### Liên hệ.

### Xem lịch sử mua hàng.

### **Giao diện quản trị*.***

### Đăng ký.

### Đăng nhập.

### Xem thông tin sản phẩm.

### Thêm mới sản phẩm

### Cập nhật sản phẩm.

### Xóa sản phẩm.

### Xem thông tin danh mục sản phẩm.

### Thêm danh mục sản phẩm.

### Sửa danh mục sản phẩm.

### Xóa danh mục.

### Xem thông tin bai viết.

### Thêm bài viết.

### Cập nhật bài viết.

### Xóa bài viết.

### Xem thông tin đơn hàng.

### Cập nhật đơn hàng.

## Đánh giá kết quả.

### **Kết quả và hướng phát triển.**

### **Hạn chế của đề tài.**

**KẾT LUẬN**

# TÀI LIỆU THAM KHẢO