

**THỰC TẬP CHUYÊN ĐỀ VÀ ĐỒ ÁN CHUYÊN MÔN**

***ĐỀ TÀI***

**<<TÊN ĐỀ TÀI>>**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:** | **Đinh Tiến Triển** |
| **Mã sinh viên:** | **2310900107** |
| **Lớp:** | **K23CNT3** |
| **Khóa:** | **2023-2027** |

**MỤC LỤC**

[DANH MỤC CÁC TỪ VIẾT TẮT 13](#_Toc186812741)

[DANH MỤC BẢNG BIỂU 19](#_Toc186812742)

[DANH MỤC HÌNH ẢNH 20](#_Toc186812743)

[CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT 21](#_Toc186812745)

[1.1. Tổng quan về lập trình thiết kế giao điện website 21](#_Toc186812746)

[1.1.1. Giới thiệu về HTML5 21](#_Toc186812747)

[1.1.2. Giới thiệu về CSS3 21](#_Toc186812748)

[1.1.3. Tổng quan về ngôn ngữ lập trình Javascript 22](#_Toc186812749)

[1.1.4. Thư viện Jquery 23](#_Toc186812750)

[1.1.5. Framework Bootstrap 24](#_Toc186812751)

[1.2. Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server 25](#_Toc186812752)

[1.2.1. Giới thiệu về SQL Server 25](#_Toc186812753)

[1.2.2. Ưu điểm của SQL Server 25](#_Toc186812754)

[1.2.3. Nhược điểm của SQL Server 26](#_Toc186812755)

[1.3. Tổng quan về ngôn ngữ lập trình phía máy chủ 26](#_Toc186812756)

[1.3.1. Giới thiệu về nền tảng công nghệ .NET 26](#_Toc186812757)

[1.3.2. Giới thiệu về ngôn ngữ lập trình C# 27](#_Toc186812758)

[1.3.3. Tổng quan về Framework ASP.NET MVC 5 28](#_Toc186812759)

[CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI 29](#_Toc186812760)

[2.1 Đặt vấn đề 29](#_Toc186812761)

[2.2 Hệ thống hiện tại 29](#_Toc186812762)

[2.3 Hệ thống đề xuất 30](#_Toc186812763)

[2.4 Giới hạn của hệ thống 30](#_Toc186812764)

[2.5 Yêu cầu về phần cứng và phần mềm 31](#_Toc186812765)

[2.5.1 Yêu cầu tối thiểu 31](#_Toc186812766)

[2.1.1 Yêu cầu đề nghị 31](#_Toc186812767)

CHƯƠNG 3: [PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG 32](#_Toc186812769)

[3.1. Người dùng của hệ thống 32](#_Toc186812770)

[3.2. Chức năng của hệ thống 33](#_Toc186812771)

[3.3. Systeme Designs (Thiết kế hệ thống) 35](#_Toc186812836)

[3.3.1. Entity Relationship Diagram (Mô hình quan hệ thực thể) 36](#_Toc186812837)

[3.4. Database Design (Thiết kế cơ sở dữ liệu) 37](#_Toc186812838)

[3.5. Table RelationShip Diagram (Mô hình quan hệ bảng) 41](#_Toc186812840)

[3.6. SiteMap (Sơ đồ cấu trúc site) 42](#_Toc186812842)

[3.6.1. Site map (Cấu trúc trang): Dành cho khách vãng lai 42](#_Toc186812843)

[3.6.2. Sitemap (Cấu trúc trang): Dành cho thành viên 42](#_Toc186812845)

[3.6.3. Sitemap (Cấu trúc trang): Dành cho quản trị viên 43](#_Toc186812847)

[3.7. Algorithms (Giải thuật) 44](#_Toc186812848)

[3.7.1. Đăng ký 44](#_Toc186812849)

[3.7.2. Đăng nhập 45](#_Toc186812851)

[3.7.3. Đăng xuất 46](#_Toc186812853)

[3.7.4. Xóa sản phẩm 46](#_Toc186812855)

[CHƯƠNG 4: XÂY DỰNG HỆ THỐNG 47](#_Toc186812857)

[4.1. Xây dựng phần mềm 47](#_Toc186812858)

[4.1.1. Một số mã nguồn chương trình 47](#_Toc186812859)

[4.1.2. Một số giao diện người dùng và chức năng cụ thể 47](#_Toc186812862)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 49](#_Toc186812865)

[5.1 Kết luận 49](#_Toc186812866)

[5.2 Task Sheet (bảng kế hoạch công việc) 50](#_Toc186812867)

[5.3 Checklist (bảng kiểm tra chức năng) 52](#_Toc186812868)

[5.3.1 Kiểm tra dữ liệu nhập 52](#_Toc186812869)

[5.3.2 Kiểm tra các liên kết 52](#_Toc186812870)

[TÀI LIỆU THAM KHẢO 53](#_Toc186812871)

**DANH MỤC CÁC TỪ VIẾT TẮT**

DANH MỤC CÁC TỪ VIẾT TẮT CHƯƠNG 1:

| Từ viết tắt | Giải thích ngắn gọn |
| --- | --- |
| HTML | Ngôn ngữ đánh dấu siêu văn bản dùng để tạo cấu trúc trang web. |
| W3C | Tổ chức chuẩn hóa các công nghệ liên quan đến web. |
| API | Giao diện lập trình ứng dụng, giúp tương tác giữa các hệ thống. |
| DOM | Mô hình cấu trúc của tài liệu HTML hoặc XML. |
| AJAX | Công nghệ tải dữ liệu không đồng bộ mà không cần tải lại trang. |
| CSS | Ngôn ngữ định kiểu để trình bày giao diện website. |
| RGBA | Định dạng màu trong CSS với hỗ trợ độ trong suốt. |
| HSLA | Định dạng màu sắc trong CSS với độ trong suốt và ánh sáng. |
| JS | Ngôn ngữ lập trình kịch bản phía client để tạo tương tác web. |
| SQL | Ngôn ngữ truy vấn dữ liệu quan hệ. |
| RDBMS | Hệ quản trị cơ sở dữ liệu quan hệ. |
| ACID | Tính chất đảm bảo của giao dịch trong cơ sở dữ liệu. |
| .NET | Nền tảng phát triển ứng dụng của Microsoft. |
| ORM | Công cụ ánh xạ giữa cơ sở dữ liệu và đối tượng lập trình. |

DANH MỤC CÁC TỪ VIẾT TẮT CHƯƠNG 2:

|  | Từ viết tắt | Giải thích ngắn gọn |
| --- | --- | --- |
|  | AI | Artificial Intelligence (Trí tuệ nhân tạo) |
|  | API | Application Programming Interface |
|  | HDD | Hard Disk Drive |
|  | CPU | Central Processing Unit |
|  | RAM | Random Access Memory |
|  | SQL | Structured Query Language |
|  | MySQL | Hệ quản trị cơ sở dữ liệu quan hệ MySQL |
|  | HTTP | Hypertext Transfer Protocol |
|  | HTTPS | Hypertext Transfer Protocol Secure |
|  | UX | User Experience (Trải nghiệm người dùng) |
|  | UI | User Interface (Giao diện người dùng) |
|  | IE | Internet Explorer |
|  | GHz | Gigahertz |
|  | GB | Gigabyte |
|  | PenIV | Intel Pentium IV |

DANH MỤC CÁC TỪ VIẾT TẮT CHƯƠNG 3:

| Từ viết tắt | Đầy đủ |
| --- | --- |
| Guest | Khách vãng lai |
| Member | Thành viên |
| Admin | Quản trị viên |
| ERD | Entity Relationship Diagram |
| SQL | Structured Query Language |
| DB | Database |
| CRUD | Create, Read, Update, Delete |
| API | Application Programming Interface |
| UI | User Interface |
| UX | User Experience |
| JSON | JavaScript Object Notation |
| PHP | Hypertext Preprocessor |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| XML | eXtensible Markup Language |
| CSS | Client-Side Scripting |
| AJAX | Asynchronous JavaScript and XML |
| REST | Representational State Transfer |
| JWT | JSON Web Token |
| SSL | Secure Sockets Layer |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| SQL Server | Microsoft SQL Server |

DANH MỤC CÁC TỪ VIẾT TẮT CHƯƠNG 4:

| Từ viết tắt | Giải thích |
| --- | --- |
| API | Giao diện lập trình ứng dụng, cho phép các ứng dụng tương tác với nhau. |
| UI | Giao diện người dùng, là phần của hệ thống mà người dùng tương tác để thực hiện các tác vụ. |
| UX | Trải nghiệm người dùng, tổng thể cảm nhận của người dùng khi sử dụng hệ thống. |
| SQL | Ngôn ngữ truy vấn cấu trúc, được sử dụng để quản lý và thao tác với cơ sở dữ liệu quan hệ. |
| PHP | Ngôn ngữ lập trình kịch bản phổ biến, chủ yếu dùng để phát triển ứng dụng web. |
| CRUD | Các thao tác cơ bản trong quản lý cơ sở dữ liệu: tạo, đọc, cập nhật và xóa dữ liệu. |
| SSL | Giao thức mã hóa bảo mật, được sử dụng để bảo vệ dữ liệu truyền tải giữa người dùng và máy chủ web. |
| HTML | Ngôn ngữ đánh dấu siêu văn bản, được sử dụng để tạo cấu trúc các trang web. |
| CSS | Bộ quy tắc định kiểu, dùng để điều chỉnh bố cục và thiết kế của trang web. |
| AJAX | Kỹ thuật phát triển web giúp trao đổi dữ liệu với máy chủ mà không làm mới lại toàn bộ trang web. |
| JWT | Chuẩn mã hóa sử dụng để truyền thông tin xác thực giữa client và server trong các ứng dụng web. |
| DB | Cơ sở dữ liệu, nơi lưu trữ thông tin hệ thống, bao gồm các bảng và các mối quan hệ giữa chúng. |
| REST | Kiến trúc phần mềm, chủ yếu dùng trong phát triển dịch vụ web, cho phép các hệ thống trao đổi dữ liệu qua HTTP. |

DANH MỤC CÁC TỪ VIẾT TẮT CHƯƠNG 5:

| Từ viết tắt | Giải thích |
| --- | --- |
| UI | Giao diện người dùng, là phần của hệ thống mà người dùng tương tác để thực hiện các tác vụ. |
| UX | Trải nghiệm người dùng, tổng thể cảm nhận của người dùng khi sử dụng hệ thống. |
| CRUD | Các thao tác cơ bản trong quản lý cơ sở dữ liệu: tạo, đọc, cập nhật và xóa dữ liệu. |
| API | Giao diện lập trình ứng dụng, cho phép các ứng dụng tương tác với nhau. |
| CSS | Bộ quy tắc định kiểu, dùng để điều chỉnh bố cục và thiết kế của trang web. |
| HTML | Ngôn ngữ đánh dấu siêu văn bản, được sử dụng để tạo cấu trúc các trang web. |
| PHP | Ngôn ngữ lập trình kịch bản phổ biến, chủ yếu dùng để phát triển ứng dụng web. |
| SQL | Ngôn ngữ truy vấn cấu trúc, được sử dụng để quản lý và thao tác với cơ sở dữ liệu quan hệ. |
| DB | Cơ sở dữ liệu, nơi lưu trữ thông tin hệ thống, bao gồm các bảng và các mối quan hệ giữa chúng. |
| SSL | Giao thức mã hóa bảo mật, được sử dụng để bảo vệ dữ liệu truyền tải giữa người dùng và máy chủ web. |
| AJAX | Kỹ thuật phát triển web giúp trao đổi dữ liệu với máy chủ mà không làm mới lại toàn bộ trang web. |
| JWT | Chuẩn mã hóa sử dụng để truyền thông tin xác thực giữa client và server trong các ứng dụng web. |
| REST | Kiến trúc phần mềm, chủ yếu dùng trong phát triển dịch vụ web, cho phép các hệ thống trao đổi dữ liệu qua HTTP. |
| RDBMS | Hệ quản trị cơ sở dữ liệu quan hệ, giúp quản lý dữ liệu được tổ chức trong các bảng và mối quan hệ giữa chúng. |

[**DANH MỤC BẢNG BIỂU**](#_heading=h.34g0dwd)

[3.3.](#_heading=h.2szc72q) Systeme Designs (Thiết kế hệ thống)

Bảng 3.3.1. Entity Relationship Diagram (Mô hình quan hệ thực thể)

Bảng 3.3.2. Bảng quản trị

Bảng 3.3.3. Bảng khách hàng

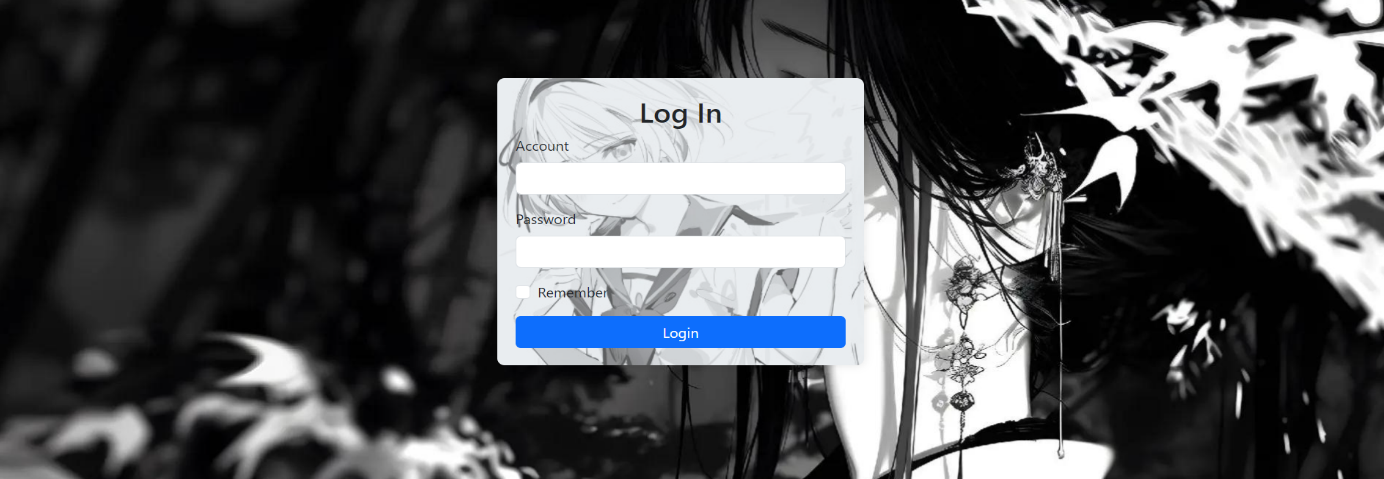
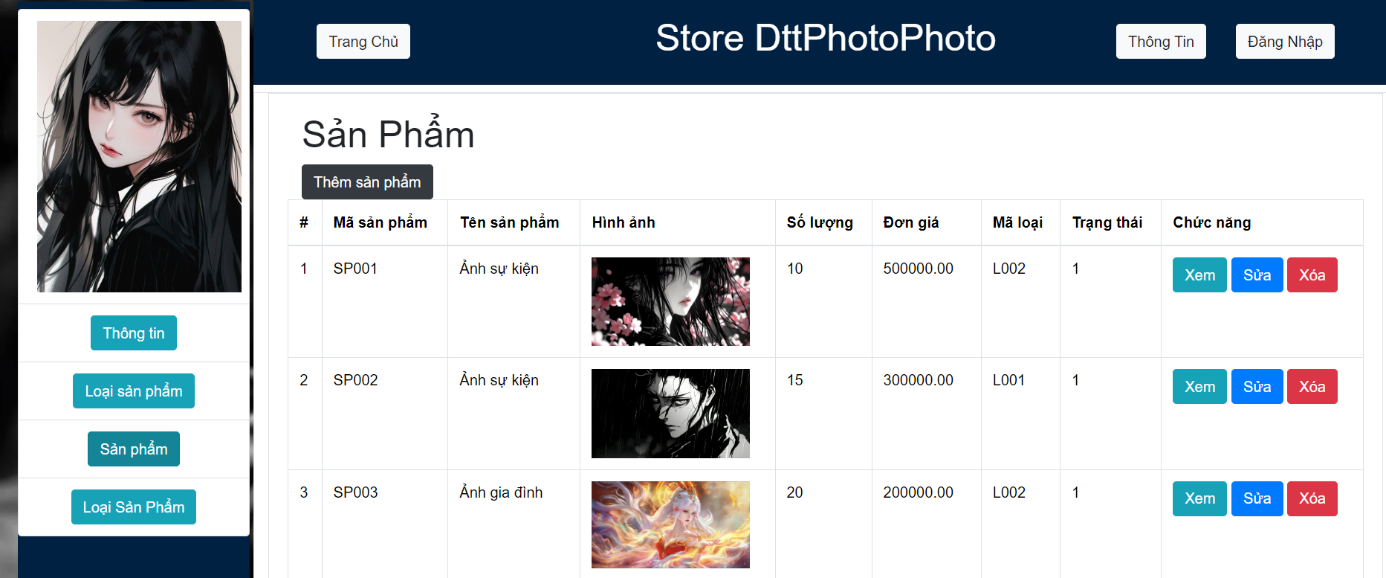
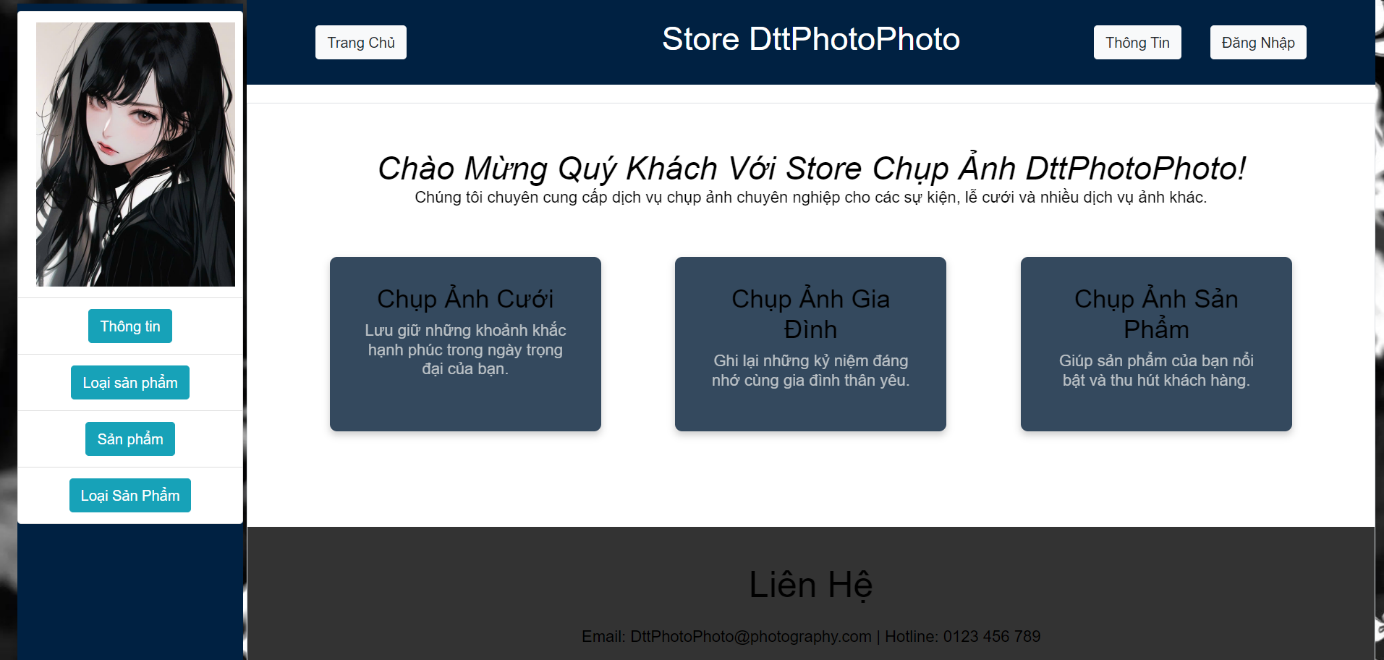
Bảng 3.3.4 Bảng loại sản phẩm

Bảng 3.3.5 Bảng sản phẩm

Bảng 3.3.6 Bảng hoá đơn

Bảng 3.3.7 Bảng chi tiết hoá đơn

**DANH MỤC HÌNH ẢNH**

****

**CHƯƠNG 1:  
NGHIÊN CỨU CƠ SỞ LÝ THUYẾT**

* 1. **Tổng quan về lập trình thiết kế giao điện website**
     1. **Giới thiệu về HTML5**

HTML5 là phiên bản mới nhất của ngôn ngữ đánh dấu HTML (HyperText Markup Language), được phát triển nhằm cải thiện khả năng xây dựng các trang web và ứng dụng web hiện đại. Được W3C chính thức ra mắt vào năm 2014, HTML5 không chỉ kế thừa các tính năng từ các phiên bản trước mà còn bổ sung nhiều cải tiến quan trọng. Điểm nổi bật của HTML5 là việc giới thiệu các thẻ semantic như <header>, <footer>, <article>, <section> và <nav>, giúp cấu trúc tài liệu rõ ràng hơn và hỗ trợ tốt hơn cho SEO cũng như các công cụ tìm kiếm.

HTML5 tích hợp các thẻ đa phương tiện như <audio> và <video>, cho phép nhúng và phát trực tiếp nội dung mà không cần sử dụng các plugin bên ngoài như Flash, giúp cải thiện hiệu suất và khả năng tương thích. Ngoài ra, HTML5 còn tăng cường khả năng tương tác thông qua các API như Canvas API để vẽ đồ họa 2D, Web Storage API để lưu trữ dữ liệu cục bộ, và Geolocation API để lấy vị trí địa lý. Đặc biệt, HTML5 được tối ưu hóa cho thiết kế Responsive, đảm bảo giao diện website hoạt động tốt trên mọi thiết bị từ desktop đến điện thoại.

Một trong những lợi ích lớn của HTML5 là tính đa nền tảng và khả năng tương thích cao với các trình duyệt hiện đại như Chrome, Firefox, Safari và Edge. Các tính năng mới như cải tiến biểu mẫu với placeholder, required hay kiểu nhập liệu đặc biệt như email, tel, date giúp việc xử lý dữ liệu trở nên chính xác và dễ dàng hơn. Tuy nhiên, HTML5 cũng gặp một số hạn chế, như thiếu sự hỗ trợ trên các trình duyệt cũ hoặc yêu cầu lập trình viên phải nắm vững kiến thức để sử dụng hiệu quả các tính năng nâng cao.

Với những cải tiến vượt bậc, HTML5 không chỉ nâng cao trải nghiệm người dùng mà còn mở ra nhiều cơ hội phát triển ứng dụng web hiện đại, trở thành tiêu chuẩn quan trọng trong lập trình giao diện website ngày nay**.**

* + 1. **Giới thiệu về CSS3**

CSS3 là phiên bản thứ ba của CSS (Cascading Style Sheets), được phát triển để định kiểu và cải thiện giao diện cho các tài liệu HTML. Ra đời nhằm đáp ứng nhu cầu ngày càng cao của thiết kế web hiện đại, CSS3 mang đến nhiều tính năng mạnh mẽ và linh hoạt, giúp tạo ra các trang web đẹp mắt và tương tác tốt hơn.

Một trong những điểm nổi bật của CSS3 là việc chia thành các mô-đun (modules), như Selectors, Box Model, Backgrounds and Borders, Text Effects, và Animations, giúp cải thiện khả năng tùy biến và tăng hiệu suất phát triển. CSS3 bổ sung nhiều thuộc tính mới, cho phép các nhà phát triển dễ dàng tạo hiệu ứng mà không cần sử dụng JavaScript hoặc hình ảnh bổ trợ.

CSS3 hỗ trợ tốt cho thiết kế Responsive, giúp giao diện trang web tự động điều chỉnh để hiển thị đẹp trên mọi kích thước màn hình, từ desktop đến smartphone. Các kỹ thuật như Media Queries, Grid Layout, và Flexbox được tích hợp mạnh mẽ, giúp tối ưu hóa bố cục và trải nghiệm người dùng. Ngoài ra, CSS3 còn hỗ trợ các hiệu ứng 2D và 3D, như transform, transition, và animation, cho phép tạo chuyển động mượt mà và thu hút.

Bên cạnh đó, CSS3 cải tiến về màu sắc và đồ họa, với sự hỗ trợ các giá trị màu sắc mới như RGBA, HSLA và gradient, giúp tạo ra các thiết kế sống động mà không cần sử dụng hình ảnh. Ngoài ra, thuộc tính như box-shadow và text-shadow cho phép thêm hiệu ứng bóng trực tiếp vào các phần tử.

Mặc dù CSS3 mang lại nhiều ưu điểm, nhưng nó cũng có một số hạn chế, như sự khác biệt trong việc hỗ trợ các tính năng giữa các trình duyệt. Điều này yêu cầu các nhà phát triển phải kiểm tra kỹ lưỡng và sử dụng giải pháp thay thế (fallback) nếu cần.

CSS3 đã trở thành tiêu chuẩn không thể thiếu trong lập trình giao diện web hiện đại. Với khả năng tùy chỉnh mạnh mẽ, dễ học và dễ sử dụng, CSS3 giúp các nhà phát triển tạo ra các thiết kế đẹp mắt, tối ưu hóa hiệu suất và nâng cao trải nghiệm người dùng.

* + 1. **Tổng quan về ngôn ngữ lập trình Javascript**

JavaScript (JS) là một ngôn ngữ lập trình phổ biến và quan trọng trong việc phát triển web hiện đại. Được tạo ra vào năm 1995 bởi Brendan Eich, JavaScript ban đầu chỉ được sử dụng để thêm các tính năng động vào các trang web trên trình duyệt. Tuy nhiên, ngày nay JavaScript đã phát triển mạnh mẽ và trở thành ngôn ngữ lập trình toàn diện, được sử dụng không chỉ ở phía client-side (frontend) mà còn ở phía server-side (backend) nhờ vào công nghệ như Node.js.

JavaScript giúp các nhà phát triển tạo ra những trang web tương tác và động bằng cách kết hợp với HTML và CSS. Trong khi HTML xác định cấu trúc trang web và CSS chịu trách nhiệm về giao diện, JavaScript đảm nhiệm việc tạo ra các hiệu ứng, xử lý sự kiện, và tương tác với người dùng. Một trong những điểm mạnh của JavaScript là khả năng hoạt động trên tất cả các trình duyệt hiện đại mà không cần cài đặt thêm phần mềm, đồng thời có cú pháp dễ học và dễ sử dụng.

JavaScript hỗ trợ nhiều tính năng mạnh mẽ như xử lý sự kiện, thao tác với DOM (Document Object Model) để thay đổi nội dung trang web, và lập trình bất đồng bộ (asynchronous) với các công cụ như Promises, Async/Await, và AJAX. Điều này cho phép các trang web tải dữ liệu từ máy chủ mà không cần tải lại toàn bộ trang, mang lại trải nghiệm người dùng mượt mà và nhanh chóng.

JavaScript còn được mở rộng qua nhiều thư viện và framework như React, Vue.js, và Angular để phát triển ứng dụng web động, cũng như Node.js và Express.js để xử lý dữ liệu phía server. Nhờ vào những tính năng này, JavaScript không chỉ được sử dụng để xây dựng các trang web tĩnh mà còn có thể phát triển các ứng dụng phức tạp, ứng dụng di động, thậm chí là các trò chơi đơn giản chạy trên trình duyệt.

Dù JavaScript có nhiều ưu điểm, ngôn ngữ này cũng gặp phải một số hạn chế như vấn đề bảo mật nếu mã không được viết đúng cách và sự khác biệt trong hỗ trợ giữa các trình duyệt, yêu cầu phải kiểm tra và tối ưu hóa kỹ lưỡng. Tuy nhiên, với sự linh hoạt, khả năng mở rộng mạnh mẽ, và cộng đồng phát triển rộng lớn, JavaScript vẫn là một trong những ngôn ngữ lập trình quan trọng và phổ biến nhất hiện nay.

* + 1. **Thư viện Jquery**

Thư viện jQuery là một thư viện JavaScript phổ biến được phát triển bởi John Resig vào năm 2006. Mục tiêu chính của jQuery là đơn giản hóa quá trình thao tác với DOM (Document Object Model), xử lý sự kiện, tạo hiệu ứng động và thực hiện AJAX, giúp mã JavaScript trở nên ngắn gọn và dễ hiểu hơn. jQuery được thiết kế để làm việc trên nhiều trình duyệt khác nhau, giúp các ứng dụng web tương thích với nhiều môi trường và giảm thiểu sự khác biệt giữa các trình duyệt.

Một số tính năng nổi bật của jQuery bao gồm thao tác DOM dễ dàng, xử lý sự kiện linh hoạt như click, hover, và submit, hỗ trợ các hiệu ứng động như fadeIn(), fadeOut(), slideUp(), slideDown(), và khả năng sử dụng AJAX để tải dữ liệu mà không cần tải lại toàn bộ trang. Điều này giúp tạo ra trải nghiệm người dùng mượt mà và tiết kiệm thời gian phát triển. Một điểm mạnh lớn của jQuery là khả năng tương thích giữa các trình duyệt, giúp các nhà phát triển tránh gặp phải vấn đề không tương thích trên các trình duyệt cũ.

jQuery cũng mang lại nhiều lợi ích cho lập trình viên, chẳng hạn như cú pháp đơn giản và dễ đọc, tiết kiệm thời gian khi thực hiện các thao tác phức tạp với DOM hay hiệu ứng, và cộng đồng hỗ trợ rộng rãi giúp dễ dàng tìm kiếm tài liệu và giải pháp cho các vấn đề gặp phải. Tuy nhiên, jQuery cũng có một số nhược điểm, như làm tăng kích thước tệp JavaScript, ảnh hưởng đến tốc độ tải trang, đặc biệt trong các ứng dụng không sử dụng nhiều tính năng của thư viện. Bên cạnh đó, trong một số trường hợp, jQuery có thể không hiệu quả bằng JavaScript thuần khi xử lý các tác vụ đơn giản.

Mặc dù hiện nay có nhiều thư viện và framework khác như React, Vue.js, và Angular có thể thay thế jQuery trong các dự án web hiện đại, nhưng thư viện này vẫn giữ vai trò quan trọng, đặc biệt trong các dự án cũ hoặc khi cần các tính năng cơ bản như thao tác DOM, hiệu ứng động, và AJAX. Với cú pháp đơn giản và khả năng tương thích đa trình duyệt, jQuery vẫn là một công cụ hữu ích cho các nhà phát triển, đặc biệt là những người mới bắt đầu học lập trình web.

* + 1. **Framework Bootstrap**

Framework Bootstrap là một công cụ mã nguồn mở phổ biến được phát triển vào năm 2011 bởi Mark Otto và Jacob Thornton tại Twitter. Bootstrap giúp việc phát triển giao diện web trở nên dễ dàng và nhanh chóng, cung cấp các thành phần giao diện người dùng có sẵn và một hệ thống lưới linh hoạt, hỗ trợ thiết kế web đáp ứng (responsive) để tối ưu hóa giao diện trên nhiều thiết bị và màn hình khác nhau. Với hệ thống lưới 12 cột, Bootstrap cho phép lập trình viên dễ dàng tạo bố cục trang web một cách nhất quán và chuyên nghiệp. Các thành phần giao diện như nút bấm, biểu mẫu, modal, cảnh báo, thanh điều hướng... đều đã được tích hợp sẵn trong Bootstrap, giúp tiết kiệm thời gian và công sức trong việc xây dựng giao diện.

Một trong những ưu điểm nổi bật của Bootstrap là tính linh hoạt và dễ tùy chỉnh. Mặc dù nó cung cấp các thành phần giao diện sẵn có, nhưng người dùng có thể dễ dàng thay đổi chúng thông qua việc thay đổi các biến cấu hình hoặc sử dụng các công cụ tùy chỉnh để phù hợp với yêu cầu riêng. Bootstrap cũng cung cấp hỗ trợ CSS và JavaScript mạnh mẽ, giúp tạo ra các hiệu ứng động như menu thả xuống, thanh cuộn ảnh và các thông báo pop-up mà không cần phải viết mã phức tạp. Ngoài ra, Bootstrap đảm bảo tính tương thích đa nền tảng, giúp các trang web hoạt động trơn tru trên nhiều trình duyệt và thiết bị khác nhau, giảm thiểu vấn đề tương thích khi phát triển ứng dụng.

Tuy nhiên, Bootstrap cũng có một số nhược điểm. Việc sử dụng các thành phần mặc định của Bootstrap có thể dẫn đến tình trạng giao diện của các trang web trở nên giống nhau, thiếu tính sáng tạo. Điều này có thể làm giảm tính độc đáo của trang web nếu không được tùy chỉnh đúng cách. Ngoài ra, việc tải toàn bộ bộ công cụ của Bootstrap, bao gồm cả CSS và JavaScript, có thể làm tăng kích thước tệp và ảnh hưởng đến tốc độ tải trang, đặc biệt khi bạn chỉ sử dụng một phần nhỏ của framework. Tuy nhiên, nếu được sử dụng hợp lý và tùy chỉnh theo nhu cầu, Bootstrap vẫn là một công cụ rất hữu ích giúp phát triển giao diện web nhanh chóng và hiệu quả.

Tóm lại, Bootstrap là một framework mạnh mẽ và tiện lợi, giúp việc phát triển giao diện web trở nên nhanh chóng và dễ dàng hơn. Với các tính năng nổi bật như thiết kế đáp ứng, hệ thống lưới linh hoạt, các thành phần giao diện sẵn có và khả năng hỗ trợ đa nền tảng, Bootstrap đã trở thành công cụ quan trọng cho các nhà phát triển web. Mặc dù có một số nhược điểm, nhưng với sự tùy chỉnh và tối ưu hóa hợp lý, Bootstrap vẫn là lựa chọn tuyệt vời cho việc xây dựng các ứng dụng web hiện đại.

* 1. **Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server**
     1. **Giới thiệu về SQL Server**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft, được sử dụng rộng rãi trong các ứng dụng doanh nghiệp và hệ thống quản lý dữ liệu. SQL Server hỗ trợ quản lý và lưu trữ dữ liệu trong các cơ sở dữ liệu, cho phép người dùng truy vấn, thêm, sửa, và xóa dữ liệu thông qua ngôn ngữ SQL (Structured Query Language). Được ra mắt lần đầu vào năm 1989, SQL Server đã phát triển và trở thành một trong những hệ quản trị cơ sở dữ liệu mạnh mẽ, phổ biến nhất hiện nay, với khả năng xử lý các khối lượng dữ liệu lớn và hỗ trợ các tính năng bảo mật và sao lưu dữ liệu mạnh mẽ. Nó có thể chạy trên các hệ điều hành Windows và Linux và tích hợp tốt với các công cụ của Microsoft như Azure, Power BI và Visual Studio. Với khả năng mở rộng linh hoạt, hỗ trợ giao dịch ACID, và tính năng bảo mật cao, SQL Server là một giải pháp lý tưởng cho các ứng dụng cần hiệu suất và độ tin cậy cao trong việc quản lý dữ liệu.

* + 1. **Ưu điểm của SQL Server**

SQL Server mang lại nhiều ưu điểm nổi bật, khiến nó trở thành lựa chọn phổ biến cho các hệ thống quản lý cơ sở dữ liệu. Đầu tiên, SQL Server có hiệu suất cao, khả năng xử lý các truy vấn phức tạp và khối lượng dữ liệu lớn một cách nhanh chóng và hiệu quả. Nó hỗ trợ các giao dịch ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính toàn vẹn và độ tin cậy cao của dữ liệu trong các môi trường yêu cầu tính nhất quán. Thêm vào đó, SQL Server có tính bảo mật mạnh mẽ, với các tính năng như mã hóa dữ liệu, phân quyền người dùng và xác thực người dùng, giúp bảo vệ dữ liệu khỏi các truy cập trái phép. SQL Server cũng cung cấp các tính năng sao lưu và phục hồi dữ liệu mạnh mẽ, giúp bảo vệ dữ liệu quan trọng và đảm bảo khả năng phục hồi khi xảy ra sự cố. Bên cạnh đó, SQL Server tích hợp tốt với các công cụ và dịch vụ của Microsoft như Azure và Power BI, giúp tăng cường khả năng phân tích và báo cáo dữ liệu. Với khả năng mở rộng linh hoạt, SQL Server có thể đáp ứng nhu cầu của các tổ chức lớn, xử lý khối lượng dữ liệu khổng lồ và hỗ trợ số lượng người dùng đồng thời cao. Tất cả những ưu điểm này giúp SQL Server trở thành một giải pháp lý tưởng cho các ứng dụng và hệ thống đòi hỏi hiệu suất cao, bảo mật và độ tin cậy.

* + 1. **Nhược điểm của SQL Server**

Mặc dù SQL Server mang lại nhiều lợi ích, nhưng nó cũng có một số nhược điểm cần lưu ý. Đầu tiên, chi phí bản quyền cao là một yếu tố đáng kể, đặc biệt là đối với các phiên bản doanh nghiệp hoặc các tính năng cao cấp. Điều này có thể tạo ra gánh nặng tài chính cho các tổ chức nhỏ hoặc các dự án có ngân sách hạn chế. Thứ hai, yêu cầu tài nguyên phần cứng lớn cũng là một nhược điểm, vì SQL Server cần một phần cứng mạnh mẽ để hoạt động tối ưu, đặc biệt khi xử lý khối lượng dữ liệu lớn hoặc hỗ trợ số lượng người dùng đồng thời cao. Điều này có thể khiến cho việc triển khai SQL Server trở nên tốn kém và khó khăn đối với các môi trường không có tài nguyên phần cứng đủ mạnh. Bên cạnh đó, việc cấu hình và tối ưu hóa SQL Server có thể khá phức tạp và đòi hỏi người quản trị phải có kiến thức chuyên sâu. Điều này làm tăng độ khó trong việc duy trì và quản lý cơ sở dữ liệu, đặc biệt đối với các tổ chức không có đội ngũ IT chuyên nghiệp. Cuối cùng, mặc dù SQL Server là một hệ quản trị cơ sở dữ liệu mạnh mẽ, nhưng trong một số trường hợp, nó có thể không linh hoạt như các giải pháp mã nguồn mở hoặc các hệ quản trị cơ sở dữ liệu khác như MySQL hoặc PostgreSQL, đặc biệt là khi yêu cầu tùy chỉnh cao hoặc yêu cầu tích hợp với các công nghệ khác không phải của Microsoft.

* 1. **Tổng quan về ngôn ngữ lập trình phía máy chủ**
     1. **Giới thiệu về nền tảng công nghệ .NET**

.NET là một nền tảng phần mềm phát triển ứng dụng được Microsoft phát triển, cung cấp các công cụ, thư viện và môi trường để xây dựng và triển khai các ứng dụng web, desktop, mobile, và các dịch vụ đám mây. Được ra mắt lần đầu tiên vào năm 2002, .NET đã trở thành một trong những nền tảng phát triển phần mềm phổ biến nhất, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ các ứng dụng doanh nghiệp đến các dịch vụ web quy mô lớn.

Một trong những điểm mạnh của nền tảng .NET là tính đa nền tảng. Trước đây, .NET chỉ hỗ trợ các hệ điều hành Windows, nhưng với sự ra đời của .NET Core, nền tảng này đã hỗ trợ phát triển ứng dụng trên các hệ điều hành khác như macOS và Linux. Điều này giúp các nhà phát triển xây dựng ứng dụng có thể chạy trên nhiều môi trường khác nhau mà không cần thay đổi mã nguồn.

.NET hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C#, VB.NET, và F#, giúp lập trình viên có thể lựa chọn ngôn ngữ phù hợp với nhu cầu và sở thích của mình. Nền tảng này cũng cung cấp một bộ công cụ phát triển mạnh mẽ, với các IDE như Visual Studio và Visual Studio Code, giúp tối ưu hóa quy trình phát triển và kiểm thử ứng dụng.

Ngoài việc phát triển các ứng dụng web và desktop, .NET còn cung cấp các thư viện và công cụ cho việc phát triển các ứng dụng di động thông qua Xamarin, và các dịch vụ đám mây với Azure. Bên cạnh đó, Entity Framework của .NET là một công cụ ORM mạnh mẽ giúp quản lý cơ sở dữ liệu một cách dễ dàng và hiệu quả.

Với những ưu điểm về tính tương thích, khả năng mở rộng và bảo mật, .NET đã trở thành một lựa chọn lý tưởng cho các nhà phát triển phần mềm, đặc biệt là đối với các tổ chức và doanh nghiệp yêu cầu các ứng dụng có hiệu suất cao, dễ bảo trì và bảo mật.

* + 1. **Giới thiệu về ngôn ngữ lập trình C#**

C# (C-Sharp) là một ngôn ngữ lập trình mạnh mẽ, đa năng được phát triển bởi Microsoft và ra mắt lần đầu tiên vào năm 2000. Được thiết kế chủ yếu cho nền tảng .NET, C# là một ngôn ngữ hướng đối tượng, an toàn về kiểu dữ liệu và dễ học, đồng thời hỗ trợ lập trình hàm và lập trình sự kiện. C# được sử dụng rộng rãi trong việc phát triển ứng dụng desktop, ứng dụng web, ứng dụng di động (thông qua Xamarin), và các dịch vụ đám mây, đặc biệt khi tích hợp với các công cụ và thư viện của .NET.

Một trong những đặc điểm nổi bật của C# là cú pháp đơn giản và dễ tiếp cận, đặc biệt đối với những lập trình viên đã quen thuộc với các ngôn ngữ như Java, C++, hoặc JavaScript. C# hỗ trợ các tính năng như quản lý bộ nhớ tự động, thông qua Garbage Collection, giúp giảm thiểu rủi ro rò rỉ bộ nhớ và cải thiện hiệu suất của ứng dụng. C# cũng hỗ trợ tính đa nền tảng thông qua .NET Core và Xamarin, cho phép phát triển ứng dụng có thể chạy trên các hệ điều hành khác nhau như Windows, macOS, và Linux.

C# có tính mạnh mẽ và bảo mật cao, với các tính năng như kiểu dữ liệu mạnh và kiểm tra lỗi tại biên dịch, giúp phát hiện lỗi sớm trong quá trình phát triển. C# còn hỗ trợ lập trình bất đồng bộ thông qua các từ khóa như async và await, giúp dễ dàng xử lý các tác vụ I/O mà không làm tắc nghẽn ứng dụng.

C# được sử dụng rộng rãi trong các ứng dụng doanh nghiệp, từ các hệ thống ERP, CRM cho đến các ứng dụng web động sử dụng ASP.NET. Ngoài ra, C# còn rất phổ biến trong phát triển game thông qua Unity, một trong những công cụ phát triển game nổi tiếng, giúp tạo ra các trò chơi đa nền tảng.

* + 1. **Tổng quan về Framework ASP.NET MVC 5**

ASP.NET MVC 5 là một framework phát triển ứng dụng web mạnh mẽ được Microsoft phát triển, sử dụng mô hình MVC (Model-View-Controller) để phân tách rõ ràng các thành phần của ứng dụng, giúp việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn. Mô hình MVC giúp quản lý dữ liệu (Model), giao diện người dùng (View) và các tương tác người dùng (Controller) một cách hiệu quả. Framework này cung cấp nhiều tính năng nổi bật như Routing thông minh, Hỗ trợ Web API, và Authentication mạnh mẽ, giúp lập trình viên xây dựng các ứng dụng web linh hoạt, dễ mở rộng và bảo mật cao. ASP.NET MVC 5 cũng hỗ trợ các công cụ như Bundling và Minification để tối ưu hóa hiệu suất ứng dụng. Mặc dù mang lại nhiều lợi ích, framework này yêu cầu lập trình viên có kiến thức vững về mô hình MVC và các công nghệ frontend như HTML, CSS và JavaScript. Tuy nhiên, với khả năng kiểm soát hoàn toàn và khả năng tích hợp với các công nghệ hiện đại, ASP.NET MVC 5 là sự lựa chọn tuyệt vời cho việc phát triển các ứng dụng web quy mô lớn và phức tạp**.**

**CHƯƠNG 2:  
TỔNG QUAN VỀ ĐỀ TÀI**

* 1. **Đặt vấn đề**

Trong bối cảnh thị trường hiện nay, các dịch vụ chụp ảnh ngày càng phát triển và đa dạng, đáp ứng nhiều nhu cầu từ cá nhân, gia đình đến tổ chức doanh nghiệp. Tuy nhiên, việc tiếp cận các dịch vụ này thường gặp một số trở ngại như thiếu thông tin chi tiết, khó so sánh giá cả, hoặc mất thời gian liên hệ và đặt lịch trực tiếp. Điều này tạo ra nhu cầu về một nền tảng trực tuyến giúp kết nối người dùng với các studio chụp ảnh một cách nhanh chóng, tiện lợi và hiệu quả.

Web store chụp ảnh là một giải pháp tiềm năng, cung cấp môi trường trực tuyến giúp các studio dễ dàng tiếp cận khách hàng, đồng thời hỗ trợ khách hàng tìm kiếm, đặt lịch và so sánh dịch vụ chụp ảnh theo nhu cầu cá nhân. Trên thực tế, thị trường đã xuất hiện một số nền tảng tương tự, nhưng phần lớn chưa đáp ứng đầy đủ các yêu cầu như giao diện thân thiện, chức năng đặt lịch linh hoạt, và tích hợp các công nghệ hiện đại như AI để gợi ý phong cách chụp ảnh phù hợp.

Sự phát triển không ngừng của công nghệ số cùng thói quen mua sắm trực tuyến ngày càng phổ biến đã mở ra cơ hội lớn cho các web store chuyên biệt trong lĩnh vực chụp ảnh. Việc nghiên cứu và triển khai một giải pháp web store chuyên dụng không chỉ giúp tối ưu hóa trải nghiệm người dùng mà còn tạo lợi thế cạnh tranh cho các doanh nghiệp trong ngành.

* 1. **Hệ thống hiện tại**

Hiện tại, các hệ thống cung cấp dịch vụ chụp ảnh trên thị trường thường bao gồm các trang web hoặc ứng dụng của từng studio hoặc nền tảng tổng hợp dịch vụ. Tuy nhiên, phần lớn các hệ thống này chưa tối ưu hóa trải nghiệm người dùng. Nhiều trang web của studio chỉ cung cấp thông tin cơ bản về dịch vụ, bảng giá và thông tin liên hệ, trong khi thiếu các tính năng tiện ích như đặt lịch trực tuyến, thanh toán online, hoặc gợi ý dịch vụ phù hợp. Đối với các nền tảng tổng hợp, dù mang đến sự lựa chọn đa dạng, chúng thường thiếu sự chuyên biệt trong lĩnh vực chụp ảnh, dẫn đến việc khách hàng khó tìm thấy các dịch vụ phù hợp với nhu cầu cá nhân. Bên cạnh đó, giao diện và tính năng của các hệ thống hiện tại thường chưa thân thiện hoặc đủ linh hoạt, đặc biệt khi người dùng muốn cá nhân hóa dịch vụ hoặc tìm kiếm phong cách chụp ảnh độc đáo. Những hạn chế này đặt ra yêu cầu cấp thiết cho việc phát triển một hệ thống web store chuyên dụng, đáp ứng tốt hơn các yêu cầu cụ thể của khách hàng và thị trường.

* 1. **Hệ thống đề xuất**

Để khắc phục những hạn chế của các hệ thống hiện tại, hệ thống web store chụp ảnh được đề xuất sẽ tập trung vào việc cung cấp một nền tảng trực tuyến toàn diện, hiện đại và thân thiện với người dùng. Hệ thống này sẽ tích hợp các tính năng chính như tìm kiếm và so sánh dịch vụ chụp ảnh từ nhiều studio, đặt lịch trực tuyến linh hoạt, quản lý thanh toán dễ dàng, và cá nhân hóa trải nghiệm thông qua các gợi ý dựa trên sở thích hoặc xu hướng của người dùng. Ngoài ra, hệ thống sẽ sử dụng công nghệ trí tuệ nhân tạo (AI) để phân tích phong cách chụp ảnh, hỗ trợ người dùng lựa chọn dịch vụ phù hợp nhất. Bên cạnh đó, giao diện trực quan và dễ sử dụng sẽ là yếu tố quan trọng, giúp cả khách hàng và các studio dễ dàng thao tác trên nền tảng. Với những tính năng ưu việt, hệ thống đề xuất không chỉ tối ưu hóa trải nghiệm người dùng mà còn hỗ trợ các studio chụp ảnh mở rộng thị trường và nâng cao hiệu quả kinh doanh.

* 1. **Giới hạn của hệ thống**

Mặc dù hệ thống web store chụp ảnh được đề xuất mang lại nhiều lợi ích, nhưng vẫn tồn tại một số giới hạn cần xem xét. Trước tiên, hệ thống phụ thuộc nhiều vào việc các studio chụp ảnh tham gia và cung cấp thông tin đầy đủ, chính xác về dịch vụ, giá cả và lịch trình. Nếu dữ liệu này không được duy trì nhất quán, trải nghiệm người dùng có thể bị ảnh hưởng. Thứ hai, khả năng cá nhân hóa dịch vụ thông qua trí tuệ nhân tạo sẽ phụ thuộc vào chất lượng dữ liệu thu thập từ người dùng, điều này đòi hỏi thời gian và sự tương tác thường xuyên từ khách hàng. Bên cạnh đó, hệ thống có thể gặp khó khăn trong việc hỗ trợ các nhu cầu đặc thù hoặc dịch vụ ngoài phạm vi thiết kế ban đầu. Cuối cùng, các vấn đề liên quan đến bảo mật thông tin cá nhân và dữ liệu giao dịch cũng là một thách thức cần được giải quyết để đảm bảo lòng tin từ phía người dùng. Những giới hạn này sẽ là trọng tâm để cải thiện trong các phiên bản nâng cấp sau này của hệ thống.

* 1. **Yêu cầu về phần cứng và phần mềm** 
     1. **Yêu cầu tối thiểu**

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram:4GB  \* HDD >=500MB free  \* CPU: PenIV 3.0 GHZ  \* Connect the Internet. | \* Ram: 4GB  \* CPU : PenIV 2.2 GHZ  \* Connect the Internet. |
| Software | \* Hệ điều hành: Windows Server 2000/Linux  \* MySQL  \* Framework:Laravel | \* Hệ điều hành:Windows XP  \* IE 8.0, Firefox 4.0, Chrome 15….. |

* + 1. **Yêu cầu đề nghị**

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram:4GB  \* HDD >=1,5GB free  \* CPU: Intel Core i7  \* Connect the Internet. | \* Ram: 8GB  \* CPU : PenIV 5.2 GHZ  \* Connect the Internet. |
| Software | \* Hệ điều hành: Windows 7,10….trở lên.  \* MySQL  \* Framework:Laravel | \* Từ Windows XP / Vista / 7 / 10… trở lên.  \* IE 11.0 hoặc Firefox 16.0 trở lên, Chrome 19.0 trở lên. |

**CHƯƠNG 3:**

**PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG**

**3.1. Người dùng của hệ thống**

***Đối tượng người dùng của hệ thống bao gồm***

* Guest (Khách vãng lai)

Guest (Khách vãng lai) là những người dùng chưa đăng ký hoặc chưa đăng nhập vào hệ thống. Họ có quyền truy cập vào một số tính năng hoặc nội dung cơ bản của hệ thống, nhưng không có quyền hạn quản lý hoặc truy cập các chức năng nâng cao. Người dùng ở chế độ khách vãng lai thường chỉ có thể xem thông tin công khai, tìm kiếm sản phẩm, hoặc tương tác với các phần không yêu cầu tài khoản hoặc quyền truy cập đặc biệt. Mặc dù không yêu cầu tài khoản, nhưng khách vãng lai có thể được yêu cầu cung cấp một số thông tin cơ bản trong trường hợp muốn thực hiện các tác vụ như mua hàng hoặc đăng ký nhận thông báo. Quá trình chuyển từ trạng thái khách vãng lai sang người dùng đã đăng ký thường đơn giản và giúp hệ thống cá nhân hóa trải nghiệm của người dùng khi họ đăng nhập hoặc tạo tài khoản.

* Member (Thành viên)

Member (Thành viên) là người dùng đã đăng ký tài khoản và có quyền truy cập vào các tính năng và dịch vụ của hệ thống, tùy thuộc vào quyền hạn được cấp. Các thành viên có thể cập nhật thông tin cá nhân, tham gia các hoạt động như đăng bài, tham gia thảo luận hoặc mua sắm, và sử dụng các dịch vụ mà khách vãng lai không thể truy cập. Quyền hạn của thành viên có thể thay đổi tùy theo vai trò trong hệ thống. Việc trở thành thành viên giúp cá nhân hóa trải nghiệm người dùng, cung cấp các dịch vụ phù hợp và cho phép hưởng các quyền lợi đặc biệt như giảm giá, quyền truy cập sớm hoặc nhận thông báo cập nhật.

* Admin (Quản trị hệ thống)

Admin (Quản trị hệ thống) là người có quyền quản lý toàn bộ hệ thống. Quản trị viên có khả năng cấu hình hệ thống, quản lý tài khoản người dùng, giám sát hoạt động và đảm bảo bảo mật dữ liệu. Họ có thể tạo, sửa đổi hoặc xóa tài khoản, phân quyền truy cập và thực hiện các tác vụ nâng cao. Quản trị viên cũng theo dõi báo cáo, khắc phục sự cố và xử lý các vấn đề kỹ thuật, đảm bảo hệ thống hoạt động ổn định và an toàn. Vai trò của họ là cực kỳ quan trọng trong việc duy trì hiệu suất và bảo mật hệ thống.

**3.2. Chức năng của hệ thống**

*3.2.1.* ***Guest  (Chức năng dành cho khách vãng lai)***

* **Xem sản phẩm**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể xem thông tin chi tiết về các dịch vụ chụp ảnh của studio, bao gồm loại dịch vụ, phong cách chụp, giá cả, đánh giá và ảnh mẫu. |
| **Nhập** | Khách hàng lựa chọn dịch vụ, phong cách chụp hoặc studio cần xem. |
| **Xử lý** | Tạm lưu thông tin vào bộ nhớ cục bộ (local storage) hoặc cơ sở dữ liệu tạm thời. |
| **Xuất** | Hiển thị thông báo đã thêm thành công vào danh sách yêu thích. |

* **Đăng ký thành viên**

|  |  |
| --- | --- |
| **Mô tả** | Khách vãng lai có thể đăng ký tài khoản để trở thành thành viên, qua đó truy cập các chức năng nâng cao như đặt lịch, lưu danh sách yêu thích hoặc quản lý thông tin cá nhân. |
| **Nhập** | Người dùng điền các thông tin cần thiết vào biểu mẫu đăng ký, bao gồm:   * Tên người dùng (username). * Email. * Mật khẩu. * Thông tin bổ sung như họ tên, số điện thoại, địa chỉ (nếu cần). |
| **Xử lý** | Hệ Thống xử lý gồm:   * Kiểm tra username, email đã được đăng ký hay chưa. * Mật khẩu phải gồm ít nhất 6 ký tự và không có ký tự trắng. * Email nhập vào phải đúng định dạng. * ... * Yêu cầu nhập các thông tin bắt buộc. * Thêm mới các thông tin khách hàng, mã hóa mật khẩu vào CSDL nếu đúng. Nếu không yêu cầu nhập lại. |
| **Xuất** | Thông báo khách hàng đăng ký thành công sau khi đã điền đầy đủ thông tin hợp lệ. Quay về trang chủ. |

* + 1. ***Admin functions*** *(Chức năng dành cho quản trị viên)*

***Quản trị viên có các chức năng chính như sau:***

* **Đăng nhập quản trị**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên đăng nhập vào hệ thống bằng cách nhập username và password vào form đăng nhập. |
| **Nhập** | Nhập tài khoản và mật khẩu của quản trị viên. |
| **Xử lý** | Kiểm tra thông tin nhập vào có hợp lệ hay không. |
| **Xuất** | Chuyển đến trang quản trị nếu đăng nhập thành công  Quay lại trang đăng nhập nếu nhập sai. |

* **Đăng xuất quản trị**

|  |  |
| --- | --- |
| **Mô tả** | Thoát khỏi tài khoản quản trị viên. |
| **Nhập** | Click vào nút thoát từ giao diện quản trị. |
| **Xử lý** | Hủy session quản trị viên đã đăng nhập. |
| **Xuất** | Quay lại trang chủ. |

* **Quản lý tài khoản khách hàng**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc khóa/kích hoạt, xóa các tài khoản của thành viên. |
| **Nhập** | Chọn lệnh tương ứng của tài khoản cần thao tác. |
| **Xử lý** | Với việc xem thông tin tài khoản:  Hiển thị thông tin của khách hàng.  Đối với việc xóa tài khoản:  Thay đổi trạng thái quyền đăng nhập website.  Khóa/kích hoạt tài khoản:  Admin có quyền khóa/kích hoạt tài khoản của khách hàng. |
| **Xuất** | Hiện thông tin đã thay đổi. |

* **Bảng loai sản phẩm**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc khóa/kích hoạt, xóa loại sản phẩm. |
| **Nhập** | Chọn lệnh tương ứng của tài khoản cần thao tác. |
| **Xử lý** | * Với việc xem thông tin loại sản phẩm: * Hiển thị thông tin của loại sản phẩm. * Đối với việc xóa loại sản phẩm: * Hủy bỏ tất cả những hoạt động liên quan đến loại sản phẩm đã xóa. * Khóa/kích hoạt loại sản phẩm: * Admin có quyền khóa/kích hoạt loại sản phẩm. |
| **Xuất** | Hiện thông tin đã thay đổi. |

* **Bảng sản phẩm**

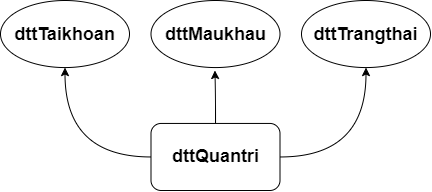
|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc khóa/kích hoạt, xóa các Sản Phẩm. |
| **Nhập** | Chọn lệnh tương ứng của sản phẩm cần thao tác. |
| **Xử lý** | * Với việc xem thông tin sản phẩm: * Hiển thị thông tin của sản phẩm. * Đối với việc xóa sản phẩm: * Hủy bỏ tất cả những hoạt động liên quan đến sản phẩm đã xóa * Khóa/kích hoạt sản phẩm: * Admin có quyền khóa/kích hoạt sản phẩm. |
| **Xuất** | Hiện thông tin đã thay đổi. |

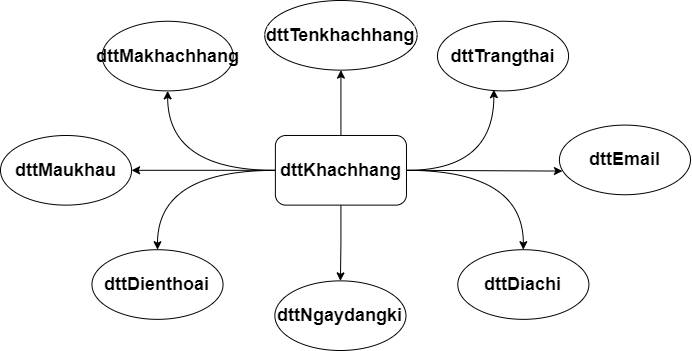
**3.3. Systeme Designs (Thiết kế hệ thống)**

Thiết kế hệ thống là quá trình xác định cấu trúc, các thành phần và tính năng của một hệ thống phần mềm, đảm bảo hệ thống đáp ứng được các yêu cầu chức năng và phi chức năng của người dùng. Quá trình này được chia thành hai giai đoạn chính: thiết kế tổng thể và thiết kế chi tiết. Trong thiết kế tổng thể, các thành phần chính của hệ thống như mô-đun, cơ sở dữ liệu, và giao diện người dùng được xác định, đồng thời định hình cách các thành phần này tương tác với nhau. Thiết kế chi tiết đi vào các yếu tố cụ thể hơn, như phương thức, chức năng, cấu trúc dữ liệu và các giao thức sử dụng trong hệ thống. Mục tiêu của thiết kế hệ thống là xây dựng một hệ thống dễ bảo trì, mở rộng, có hiệu suất và bảo mật tối ưu. Thiết kế hệ thống đóng vai trò quan trọng trong việc tạo ra nền tảng vững chắc cho quá trình phát triển và triển khai phần mềm.

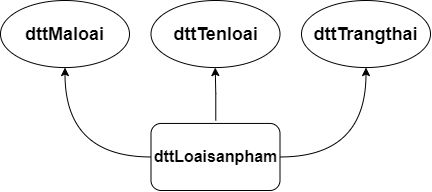
**3.3.1. Entity Relationship Diagram (Mô hình quan hệ thực thể)**

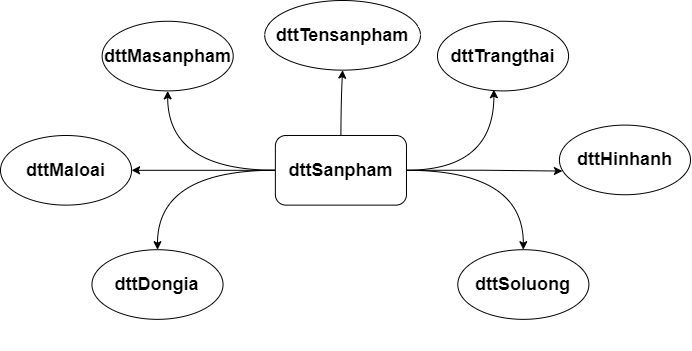
***Mô tả chi tiết các thuộc tính của các thực thể***:

1. Thực thể quản trị (**dttQuantri**):   
   Lưu trữ thông tin người quản trị viên hệ thống. 
2. Thực thể khách hàng (**dttKhachhang**):

 Lưu trữ thông tin khách hàng khi đăng ký, mua hàng tại hệ thống.

1. Thực thể khách hàng(**dttLoaisanpham):**



1. Thực thể khách hàng (**dttSanpham):** ****

**3.4. Database Design (Thiết kế cơ sở dữ liệu)**

**Thiết kế có sở dữ liệu trên hệ quản trị dữ liệu MS SQL (SQL Server)**

1. Bảng **dttquantri**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **ID** | Bigint(20) | No | Primary Key,  Auto\_increment |  |
| dttTaikhoan | Varchar(255) | No | UNIQUE | Tài khoản đăng nhập của quản trị |
| dttMatkhau | Varchar(255) | No |  | Mật khẩu đăng nhập của quản trị. |
| dttTrangthai | Tinyint(4) |  |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

1. Bảng **dttKhachhang**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **ID** | Bigint(20) | No | Primary Key; Auto\_increment |  |
| dttMakhachhang | Varchar(255) | No | Unique | Mã khách hàng |
| dttHoTenkhachhang | Varchar(255) | No |  | Tên khách hàng |
| dttTaikhoan | Varchar(255) | No | Unique | Tài khoản đăng nhập |
| dttMatkhau | Varchar(255) | No |  | Mật khẩu |
| dttDiachi | Varchar(255) | No |  | Địa chỉ |
| dttDienthoai | Varchar(10) | No | Unique | Điện thoại |
| dttEmail | Varchar(255) | No | Unique | Hộp thư điện tử |
| dttNgaydangky | DateTime | Yes | Current | Ngày cập nhật vào hệ thống |
| dttTrangthai | Tinyint(4) | Yes |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

1. Bảng **dttloaisanpham**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **ID** | Bigint(20) | No | Primary Key; Auto\_increment |  |
| dttMaloai | Varchar(25) | No | Unique | Mã loại sản phẩm |
| dttTenloai | Varchar(255) | No |  | Tên loại sản phẩm |
| dttTrangthai | Tinyint(4) | Yes |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

1. Bảng **dttsanpham**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **ID** | Bigint(20) | No | Primary Key; Auto\_increment |  |
| dttMasanpham | Varchar(255) | No | Unique | Mã sản phẩm |
| dttTensanpham | Varchar(255) | No |  | Tên sản phẩm |
| dttHinhanh | Varchar(255) | Yes |  | Hình ảnh |
| dttSoluong | Int(11) | No |  | Số lượng |
| dttDongia | Decimal(15,2) | No |  | Đơn giá |
| dttMaloai | Varchar(255) | No |  | Mã loại sản phẩm |
| dttTrangthai | Tinyint(4) | No |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

1. Bảng **dtthoadon**

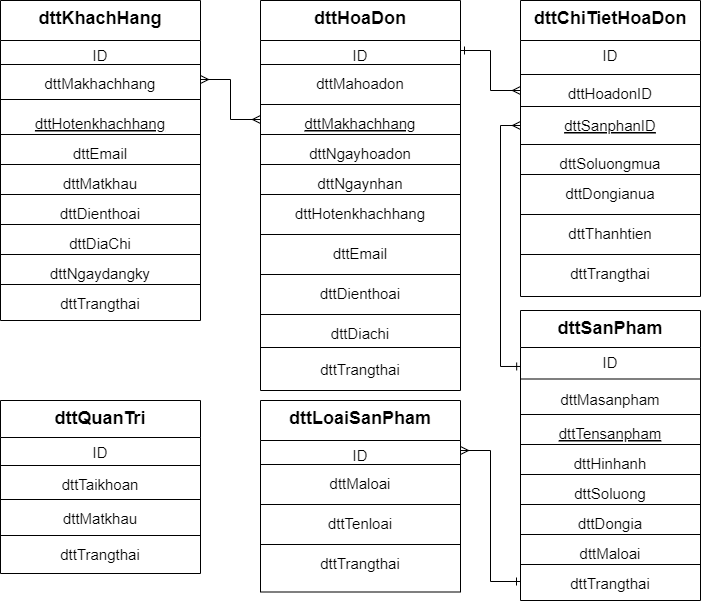
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **ID** | Bigint(20) | No | Primary Key; Auto\_increment |  |
| dttMahoadon | Varchar(255) | No | Unique | Mã hóa đơn |
| dttMakhachhang | Varchar(255) | No | Unique | Mã khách hàng |
| dttNgayhoadon | Datetime | No | Default(now) | Ngày hóa đơn |
| dttNgaynhan | Datetime | No |  | Ngày nhận |
| dttHotenkhachhang | Varchar(255) | No |  | Tên khách hàng |
| dttEmail | Varchar(255) | No | Unique | Hộp thư điện tử |
| dttDienthoai | Varchar(255) | No | Unique | Điện thoại |
| dttDiachi | Varchar(255) | No |  | Hộp thư điện tử |
| dttTongtrigia | Double(8,2) | No |  | Giá trị |
| dttTrangthai | Tinyint(4) | No | Default(1) | Trạng thái:  1- Đã xử lý 0- Đang xử lý |

1. Bảng **dttchitiethoadon**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **ID** | Bigint(20) | No | Primary Key; Auto\_increment |  |
| dttHoadonID | Bigint(20) | No | Unique | Mã hóa đơn |
| dttSanphamID | Bigint(20) | No | Unique | Mã sản phẩm |
| dttSoluongmua | Int(11) | No |  | Số lượng mua |
| dttDongiamua | Double(8,2) | No |  | Giá trị sản phẩm |
| dttThanhtien | Double(8,2) | No |  | Tổng tiền |
| dttTrangthai | Tinyint(4) | No | Default(1) | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

Thiết kế cơ sở dữ liệu là quá trình xây dựng cấu trúc dữ liệu và cách thức lưu trữ, quản lý trong một hệ thống cơ sở dữ liệu, nhằm đảm bảo hệ thống có thể đáp ứng các yêu cầu về chức năng và hiệu suất. Quá trình này bắt đầu bằng việc phân tích yêu cầu của người dùng để xác định các thực thể và mối quan hệ cần lưu trữ. Tiếp theo, mô hình khái niệm được xây dựng, thông qua các biểu đồ quan hệ thực thể (ERD), giúp hình dung cấu trúc dữ liệu tổng thể. Sau đó, mô hình logic sẽ chuyển đổi các thực thể và mối quan hệ thành các bảng và mối quan hệ giữa các bảng, trước khi đến giai đoạn mô hình vật lý, nơi tối ưu hóa việc lưu trữ và truy xuất dữ liệu trên phần cứng. Mục tiêu cuối cùng của thiết kế cơ sở dữ liệu là đảm bảo hệ thống hoạt động hiệu quả, bảo mật và dễ bảo trì, đồng thời tối ưu hóa hiệu suất và bảo vệ tính toàn vẹn của dữ liệu.

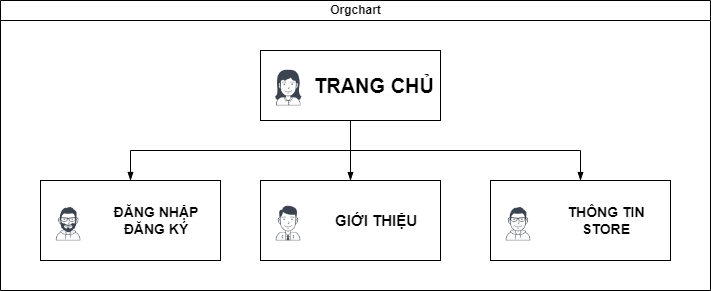
**3.5. Table RelationShip Diagram (Mô hình quan hệ bảng)**

****

Mô hình quan hệ bảng (Table Relationship Diagram) là một công cụ quan trọng trong thiết kế cơ sở dữ liệu, giúp mô tả cách các bảng dữ liệu trong hệ thống liên kết với nhau. Mô hình này thể hiện mối quan hệ giữa các bảng thông qua các khóa chính và khóa ngoại. Mỗi bảng chứa dữ liệu của một thực thể cụ thể, chẳng hạn như thông tin khách hàng, đơn hàng hay sản phẩm. Khóa chính là thuộc tính duy nhất xác định một bản ghi trong bảng, trong khi khóa ngoại được sử dụng để thiết lập mối quan hệ giữa các bảng, đảm bảo tính toàn vẹn dữ liệu. Các mối quan hệ giữa bảng có thể là một-một, một-nhiều, hoặc nhiều-nhiều, giúp kết nối và liên kết dữ liệu trong hệ thống. Mô hình quan hệ bảng giúp các nhà phát triển dễ dàng hình dung cấu trúc cơ sở dữ liệu, tối ưu hóa các truy vấn và đảm bảo tính nhất quán của dữ liệu.

**3.6. SiteMap (Sơ đồ cấu trúc site)**

**3.6.1. Site map (Cấu trúc trang): Dành cho khách vãng lai**

****

Sơ đồ cấu trúc trang (Site Map) dành cho khách vãng lai là một công cụ quan trọng để giúp người dùng nhanh chóng hiểu được cách tổ chức và phân bố các trang trên website. Đối với khách vãng lai, những người chỉ mới tiếp cận website lần đầu tiên, sơ đồ cấu trúc trang cung cấp cái nhìn tổng quan về tất cả các mục, trang và nội dung có sẵn trên website. Sơ đồ này có thể được trình bày dưới dạng danh sách hoặc đồ họa, thể hiện rõ ràng các trang chính, các trang con và liên kết giữa chúng. Việc có một sơ đồ cấu trúc rõ ràng giúp khách vãng lai dễ dàng tìm kiếm thông tin, điều hướng trang web mà không gặp khó khăn. Bằng cách này, sơ đồ không chỉ hỗ trợ việc duyệt website nhanh chóng mà còn cải thiện trải nghiệm người dùng, giúp khách vãng lai không cảm thấy bị lạc lõng khi chưa quen với giao diện và cấu trúc của trang web.

**3.6.2. Sitemap (Cấu trúc trang): Dành cho thành viên**

Trang chủ

Sản phẩm

Nhãn hiệu

Tìm kiếm

Giới thiệu

Liên hệ

Hướng dẫn

Đăng xuất

Thông tin cá nhân

Giỏ hàng

Đặt mua

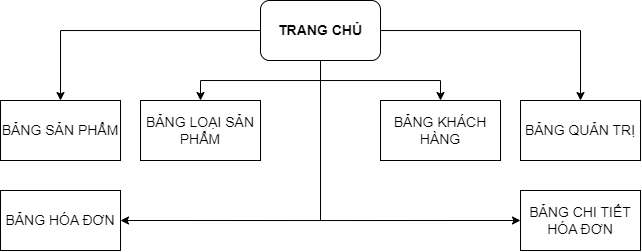
Đổi mật khẩu

Sửa thông tin

Bình luận

Sơ đồ cấu trúc trang (SiteMap) dành cho thành viên là một công cụ hữu ích giúp người dùng đã đăng ký tài khoản hoặc là thành viên của một website dễ dàng điều hướng và tìm kiếm các nội dung mà họ quan tâm. Khi người dùng đã là thành viên, họ thường sẽ có quyền truy cập vào các khu vực, trang hoặc tính năng đặc biệt mà khách vãng lai không có. Sơ đồ cấu trúc trang dành cho thành viên sẽ thể hiện không chỉ các trang chính và các trang con mà còn bao gồm các mục, dịch vụ, hay tính năng riêng biệt mà chỉ thành viên mới có thể sử dụng, như trang quản lý tài khoản, lịch sử đơn hàng, cài đặt quyền riêng tư, thông báo cá nhân, và các tài nguyên đặc biệt. Việc có một sơ đồ cấu trúc trang rõ ràng sẽ giúp thành viên dễ dàng tìm thấy những phần cần thiết, từ đó cải thiện trải nghiệm người dùng và tăng sự hài lòng khi sử dụng website.

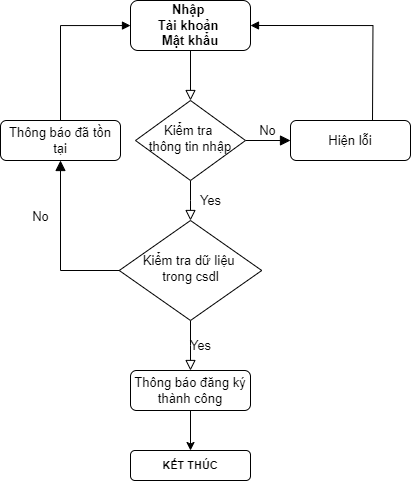
**3.6.3. Sitemap (Cấu trúc trang): Dành cho quản trị viên**



Sơ đồ cấu trúc trang (SiteMap) dành cho quản trị viên là công cụ quan trọng để hỗ trợ người quản trị website trong việc quản lý, điều hướng và kiểm soát các phần khác nhau của trang web. Đối với quản trị viên, sơ đồ cấu trúc trang sẽ bao gồm tất cả các trang và tính năng liên quan đến việc quản lý nội dung, người dùng và các cài đặt hệ thống. Điều này có thể bao gồm các mục như bảng điều khiển quản trị, quản lý người dùng, quản lý nội dung (bài viết, sản phẩm, hình ảnh, video), cài đặt hệ thống, báo cáo phân tích, và các công cụ bảo mật. Một sơ đồ cấu trúc trang dành cho quản trị viên giúp dễ dàng xác định các phần cần quản lý, từ đó nâng cao hiệu quả công việc và bảo đảm sự vận hành trơn tru của hệ thống. Việc có một cấu trúc rõ ràng và tổ chức hợp lý giúp các quản trị viên làm việc hiệu quả hơn, dễ dàng kiểm soát mọi thay đổi và cập nhật trên website mà không bị lạc trong các mục không cần thiết.

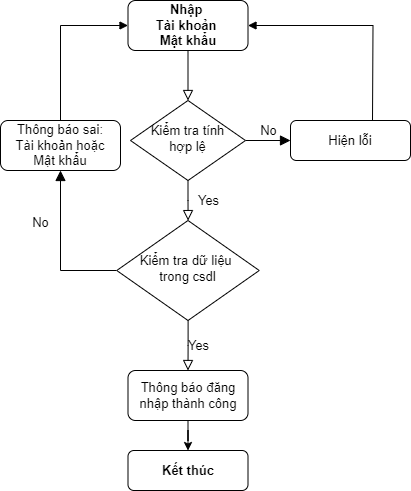
**3.7. Algorithms (Giải thuật)**

**3.7.1. Đăng ký**

****

Đăng ký là quá trình người dùng tạo tài khoản mới trên một hệ thống hoặc ứng dụng trực tuyến, cho phép họ truy cập và sử dụng các dịch vụ hoặc tính năng của hệ thống. Quá trình đăng ký thường yêu cầu người dùng cung cấp thông tin cơ bản như tên, email, mật khẩu và đôi khi là các thông tin bổ sung khác tùy vào tính chất của dịch vụ. Sau khi đăng ký thành công, người dùng sẽ nhận được thông tin xác nhận và có thể đăng nhập để sử dụng tài khoản của mình. Đăng ký không chỉ là cách để quản lý người dùng mà còn giúp bảo mật thông tin cá nhân và cho phép các nhà phát triển cung cấp các tính năng cá nhân hóa cho người dùng.

**3.7.2. Đăng nhập**

****

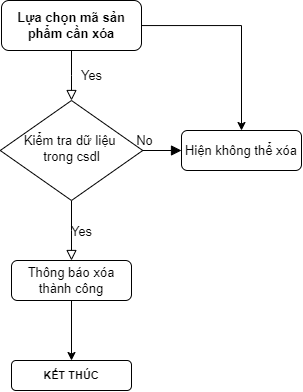
Đăng nhập là quá trình người dùng truy cập vào hệ thống hoặc ứng dụng sau khi đã có tài khoản. Để đăng nhập, người dùng thường cần cung cấp thông tin xác thực, phổ biến nhất là tên đăng nhập (hoặc email) và mật khẩu. Khi thông tin được nhập chính xác, hệ thống sẽ cho phép người dùng truy cập vào tài khoản của mình và sử dụng các dịch vụ hoặc tính năng mà họ đã đăng ký. Quá trình đăng nhập giúp bảo vệ tài khoản người dùng khỏi truy cập trái phép, đồng thời cho phép cá nhân hóa trải nghiệm người dùng.

**3.7.3. Đăng xuất**



Đăng xuất là quá trình người dùng thoát khỏi tài khoản của mình trên một hệ thống hoặc ứng dụng. Khi đăng xuất, người dùng sẽ bị ngắt kết nối khỏi tài khoản của mình, và hệ thống sẽ không còn nhận diện họ như là người dùng đã đăng nhập.

**3.7.4. Xóa sản phẩm**

****

**CHƯƠNG 4:  
XÂY DỰNG HỆ THỐNG**

**4.1. Xây dựng phần mềm**

Xây dựng phần mềm là một quá trình gồm nhiều giai đoạn, bắt đầu từ phân tích yêu cầu, thiết kế hệ thống, phát triển mã nguồn, kiểm thử, triển khai và bảo trì. Đầu tiên, nhóm phát triển cần thu thập và tài liệu hóa các yêu cầu từ khách hàng để làm cơ sở cho việc thiết kế. Sau đó, hệ thống được thiết kế với kiến trúc rõ ràng và chi tiết hóa từng module. Giai đoạn phát triển phần mềm tập trung vào lập trình, kiểm tra từng phần nhỏ để đảm bảo chất lượng. Tiếp theo, phần mềm được kiểm thử toàn diện trước khi triển khai trên môi trường thực tế. Cuối cùng, bảo trì và nâng cấp đảm bảo phần mềm luôn đáp ứng tốt nhu cầu người dùng và thích nghi với các thay đổi trong tương lai.

**4.1.1. Một số mã nguồn chương trình**

## **Các công cụ hỗ trợ cần thiết**

* + **Laravel Extension Pack :**Gói tích hợp nhiều extension hữu ích cho Laravel.
  + **Laravel Extra IntelliSense:** Cung cấp IntelliSense cho các phương thức và hàm của Laravel.
  + **Laravel Artisan:** Cho phép chạy các lệnh Artisan trực tiếp từ VS Code.
  + **PHP Debug:** Hỗ trợ gỡ lỗi cho mã PHP.
  + **Prettier**: Tự động định dạng mã để giữ cho mã sạch sẽ và dễ đọc.

## **Các công cụ phát triển cần thiết**

* + **Xampp** :cài đặt môi trường sever ảo để chạy ứng dụng Laravel trên máy tính cá nhân.
  + **Composer :** trình quản lý gói cho PHP ,cần thiết để cài đặt Laravel và các package khác.
  + **mySQL:** hệ quản trị csdl để lưu trữ dữ liệu ứng dụng.
  + **Git:** Hệ thống quản lý,lưu trữ mã nguồn.

**4.1.2. Một số giao diện người dùng và chức năng cụ thể**

**1. Giao diện trang chủ**

Giao diện trang chủ là nơi đầu tiên người dùng tương tác khi truy cập vào ứng dụng, được thiết kế để tạo ấn tượng ban đầu và cung cấp các tùy chọn điều hướng rõ ràng. Một giao diện trang chủ hiệu quả thường bao gồm tiêu đề chào mừng nổi bật, mô tả ngắn gọn về ứng dụng, và các nút chức năng như quản lý danh bạ, bán hàng, tìm kiếm hoặc thoát ứng dụng. Các thành phần được bố trí khoa học, dễ sử dụng, và kết hợp màu sắc hài hòa nhằm tăng tính thẩm mỹ và trải nghiệm người dùng. Giao diện này có thể mở rộng thêm với hình ảnh, biểu tượng hoặc tích hợp các tính năng phức tạp hơn để phù hợp với mục đích của ứng dụng.

**2. Giao diện trang quản trị**

Giao diện trang quản trị là một phần quan trọng của hệ thống, cung cấp quyền truy cập và quản lý toàn diện cho người quản trị. Giao diện này thường được thiết kế với các chức năng chính như quản lý người dùng, giám sát hệ thống, quản lý nội dung, và cấu hình cài đặt. Các nút điều hướng được sắp xếp khoa học, trực quan, kết hợp với màu sắc và phông chữ rõ ràng, tạo cảm giác chuyên nghiệp. Người quản trị có thể thực hiện các tác vụ như thêm, sửa, xóa dữ liệu, phân quyền người dùng hoặc theo dõi nhật ký hệ thống chỉ với vài thao tác đơn giản. Đây là trung tâm điều phối, đảm bảo hệ thống hoạt động ổn định và hiệu quả.

**CHƯƠNG 5:  
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1 Kết luận**

Quá trình làm đồ án chuyên đề với đề tài "……" đã giúp em có cơ hội để trau dồi các kiến thức được học từ thầy cô trên giảng đường, đồng thời cũng tiếp thu và tìm hiểu được thêm nhiều kiến thức, kinh nghiệm mới, cụ thể là:

* + - Nắm được các quy trình nghiệp vụ trong phân tích và thiết kế một sản phẩm phần mềm.
    - Hiểu biết được một số phương pháp phân tích thiết kế hệ thống, từ đó đã áp dụng các hiểu biết của mình để phân tích thiết kế hệ thống bài toán theo mô hình ba lớp.
    - Tiếp thu được các kinh nghiệm thực tế khi thực hành xây dựng một sản phẩm cụ thể, áp dụng được các kiến thức đã học vào thực tiễn.
    - Củng cố các kiến thức về ngôn ngữ lập trình cũng như các thư viện – framework,…

**Ưu điểm:**

Hệ thống cơ bản đã hoàn thiện đầy đủ các tính năng đã được thiết kế và đưa vào sử dụng phục vụ các yêu cầu của người dùng. Quan trọng hơn, hệ thống đảm bảo duy trì được sự ổn định, tính bảo mật và tính tiện dụng cho người sử dụng. Các tính năng đã hoàn thành:

* + - Thêm mới sửa xóa quản trị
    - Thêm mới sửa xóa loại sản phẩm
    - Thêm mới sửa xóa sản phẩm
    - Thêm mới khách hàng
    - Trang chủ dành cho khách vãng lai

**Nhược điểm:**

Trong những phiên bản tiếp theo, ứng dụng có thể được mở rộng để đáp ứng nhiều hơn những yêu cầu từ người sử dụng như:

* + - Giao diện của hệ thống do bản thân tự thiết kế nên vẫn chưa được hoàn thiện và đẹp mắt nhất.
    - Nhiều chức năng hoạt động chưa được mượt mà tạo hiệu ứng sử dụng không tốt với người dùng.
    - Trong thời gian tới em sẽ tiếp tục hoàn thiện các chức năng hiện thời và bổ sung thêm các chức năng mới nhằm mang lại sự tiện dụng nhất cho người dùng.

**Hướng phát triển:**

Trong những phiên bản tiếp theo, ứng dụng có thể được mở rộng để đáp ứng nhiều hơn những yêu cầu từ người sử dụng như:

* + - Cải thiện, bảo trì những lỗi phát sinh ở các tính năng hiện tại.
    - Cải thiện giao diện thân thiện hơn với người dùng.
    - Phát triển thêm tính năng bình luận của khách hàng,liên hệ,tích hợp thêm để có thể trở thành 1 trang web hoàn chỉnh với đầy đủ loại vai trò.

## **Task Sheet (bảng kế hoạch công việc)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Giai đoạn** | **Từ ngày … đến….** | **Công việc** | **Tỉ lệ hoàn thành** |
| Chương 1:  (100%) | 18/12/2024-24/12/2024 | Nghiên cứu tài liệu lý thuyết cho việc sử dụng tài nguyên để xây dựng một trang web phù hợp yêu cầu của khách hàng  Lựa chọn các framework phù hợp máy tính và tìm hiểu chức năng,điểm mới,khác biệt so với các framework khác | 100% |
| Chương 2:  (100%) | 24/12/2024-  25/12/2024 | Tìm hiểu về cấu hình chung của đa số các thiết bị hiện nay để tạo ra thành phẩm thích hợp với đa số các thiết bị người dùng hiện nay | 100% |
| Chương 3:  (100%) | 25/12/2024-28/12/2024 | Tìm hiểu và xây dựng các chương trình đầu tiên,nghiên cứu về các loại giao diện để phù hợp với yêu cầu của khách hàng và dễ dàng thao tác.  Tạo cơ sở dữ liệu migation để dễ quản trị  Thiết kế kết các mối quan hệ hợp lí cân đối | 100% |
| Chương 4:  (100%) | 28/12/2024-29/12/2024 | Tìm hiểu thêm về một số thông tin,chức năng về các framework để có thể hoàn thiện lại hệ thống,tìm hiểu thêm chức năng và chỉnh sửa lại một số vấn đề để đảm bảo tính ổn định của hệ thống  Thêm css cho đầy tính nổi bật | 100% |
| Chương 5:  (100%) | 30/12/2024-31/12/2024 | Hoàn thiện và kiểm tra đảm bảo trong quá trình sử dụng không phát sinh ra lỗi,bẫy lỗi các trường hợp và chuẩn bị những hướng dẫn trước khi đưa tới tay khách hàng.rút ra các bài học quan trọng nhằm hoàn thiện bản thân và cải thiện trình độ để có thể tạo ra một thành phẩm tốt hơn trong tương lai. | 100% |

## **Checklist (bảng kiểm tra chức năng)**

## **Kiểm tra dữ liệu nhập**

|  |  |  |
| --- | --- | --- |
| **TT** | **Các thành phần** | **Tình trạng** |
| 1 | Các trường trên form không được trống theo yêu cầu | OK |
| 2 | Dữ liệu nhập phù hợp (Chuỗi, số, ngày, …) | OK |
| 3 | Tài khoản đăng nhập, email là duy nhất | OK |
| 4 | Các ràng buộc (thêm, sửa, xóa) | OK |
| 5 | Kiểm tra dữ liệu nhập trên form đầy đủ | OK |
| 6 | Xử lý dữ liệu trên giỏ hàng đầy đủ | OK |

## **Kiểm tra các liên kết**

|  |  |  |
| --- | --- | --- |
| **TT** | **Các liên kết** | **Tình trạng** |
| 1 | Tất cả các trang đều hoạt động | OK |
| 2 | Tất cả các liên kết đều hoạt động đúng | OK |
| 3 | Các trang đều không có lỗi chính tả | OK |
| 4 | Các mẫu sử dụng trên mỗi trang đều phù hợp | OK |
| 5 | Các thành phần trong trang đều hoạt động | OK |
| 6 | Ứng dụng tương thích với nhiều trình duyệt phổ biến | OK |
| 7 | Dữ liệu sau đều có nhắc nhở, thông báo | OK |
| 8 | Các trang đều có liên kết đến trang khác | OK |

**TÀI LIỆU THAM KHẢO**

**Tài liệu:**

1. <https://topdev.vn/blog/backend-developer-la-gi-lo-trinh-tro-thanh-backend-developer>
2. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

**Website:**

1. https://www.w3schools.com/
2. <https://topdev.vn/blog/sql-server-la-gi/>
3. https://github.com/Trien224/K23CNT3\_DinhTienTrien\_Project1