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

***ĐỀ TÀI***

**FAST FOOD WEBSITE**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:** | **Nguyễn Văn Khánh** |
| **Mã sinh viên:** | **2310900047** |
| **Lớp:** | **K23CNT3** |
| **Khóa:** | **K23** |

**@FITNTU-PJ** 12/2024

**MỤC LỤC**

**[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ](#_heading=h.kgcv8k)** [5](#_heading=h.kgcv8k)

**[DANH MỤC BẢNG BIỂU](#_heading=h.34g0dwd)** [8](#_heading=h.34g0dwd)

**[DANH MỤC HÌNH ẢNH](#_heading=h.1jlao46)** [10](#_heading=h.1jlao46)

**[CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT](#_heading=h.43ky6rz)** [11](#_heading=h.43ky6rz)

[1.1.](#_heading=h.2iq8gzs) Tổng quan về lập trình thiết kế giao diện website 11

[1.1.1.](#_heading=h.xvir7l) Giới thiệu về HTML 11

[1.1.2.](#_heading=h.3hv69ve) Giới thiệu về CSS3 11

[1.1.3.](#_heading=h.1x0gk37) Tổng quan về ngôn ngữ lập trình Javascript 11

[1.1.4.](#_heading=h.4h042r0) Thư viện Jquery 11

[1.1.5.](#_heading=h.2w5ecyt) Framework Bootstrap 11

[1.2.](#_heading=h.1baon6m) Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server 11

[1.2.1.](#_heading=h.3vac5uf) Giới thiệu về SQL Server 11

[1.2.2.](#_heading=h.2afmg28) Ưu điểm của SQL Server 11

[1.2.3.](#_heading=h.pkwqa1) Nhược điểm của SQL Server 11

[1.3.](#_heading=h.39kk8xu) Tổng quan về ngôn ngữ lập trình phía máy chủ 11

[1.3.1.](#_heading=h.1opuj5n) Giới thiệu về nền tảng công nghệ .NET 11

[1.3.2.](#_heading=h.48pi1tg) Giới thiệu về ngôn ngữ lập trình C# 11

[1.3.3.](#_heading=h.2nusc19) Tổng quan về Framework ASP.NET MVC 5 12

**[CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI](#_heading=h.1302m92)** [13](#_heading=h.1302m92)

[2.1.](#_heading=h.3mzq4wv) Đặt vấn đề 13

[2.2.](#_heading=h.2250f4o) Hệ thống hiện tại 13

[2.3.](#_heading=h.haapch) Hệ thống đề xuất 13

[2.4.](#_heading=h.319y80a) Giới hạn của hệ thống 13

[2.5.](#_heading=h.1gf8i83) Yêu cầu về phần cứng và phần mềm 13

[2.5.1.](#_heading=h.40ew0vw) Yêu cầu tối thiểu 13

[2.5.2.](#_heading=h.2fk6b3p) Yêu cầu đề nghị 13

**[CHƯƠNG 3:](#_heading=h.upglbi)** [15](#_heading=h.upglbi)

**[PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG](#_heading=h.3ep43zb)** [15](#_heading=h.3ep43zb)

[3.1.](#_heading=h.1tuee74) Người dùng của hệ thống 15

[3.2.](#_heading=h.4du1wux) Chức năng của hệ thống 15

[3.3.](#_heading=h.2szc72q) Systeme Designs (Thiết kế hệ thống) 18

[3.3.1.](#_heading=h.184mhaj) Entity Relationship Diagram (Mô hình quan hệ thực thể) 18

[3.4.](#_heading=h.3s49zyc) Database Design (Thiết kế cơ sở dữ liệu) 18

[3.5.](#_heading=h.279ka65) Table RelationShip Diagram (Mô hình quan hệ bảng) 19

[3.6.](#_heading=h.meukdy) SiteMap (Sơ đồ cấu trúc site) 19

[3.6.1.](#_heading=h.36ei31r) Site map (Cấu trúc trang): Dành cho khách vãng lai 20

[3.6.2.](#_heading=h.1ljsd9k) Sitemap (Cấu trúc trang): Dành cho thành viên 20

[3.6.3.](#_heading=h.45jfvxd) Sitemap (Cấu trúc trang): Dành cho quản trị viên 20

[3.7.](#_heading=h.2koq656) Algorithms (Giải thuật) 21

[3.7.1.](#_heading=h.zu0gcz) Đăng ký 21

[3.7.2.](#_heading=h.3jtnz0s) Đăng nhập 22

[3.7.3.](#_heading=h.1yyy98l) Đăng xuất 22

[3.7.4.](#_heading=h.4iylrwe) Đổi mật khẩu 23

[3.7.5.](#_heading=h.2y3w247) Thay đổi thông tin cá nhân 24

[3.7.6.](#_heading=h.1d96cc0) Thêm sản phẩm 25

[3.7.7.](#_heading=h.3x8tuzt) Xóa sản phẩm 26

[3.7.8.](#_heading=h.2ce457m) Đặt mua 27

[3.7.9.](#_heading=h.rjefff) Tìm kiếm 28

**[CHƯƠNG 4: XÂY DỰNG HỆ THỐNG](#_heading=h.3bj1y38)** [29](#_heading=h.3bj1y38)

[4.1.](#_heading=h.1qoc8b1) Xây dựng phần mềm 29

[4.1.1.](#_heading=h.4anzqyu) Một số mã nguồn chương trình 29

[4.1.2.](#_heading=h.2pta16n) Một số giao diện người dùng và chức năng cụ thể 29

[1.](#_heading=h.14ykbeg) Giao diện trang chủ 29

[2.](#_heading=h.3oy7u29) Giao diện trang quản trị 29

[3.](#_heading=h.243i4a2) ….. 29

**[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN](#_heading=h.j8sehv)** [30](#_heading=h.j8sehv)

[5.1.](#_heading=h.338fx5o) Kết luận 30

[5.2.](#_heading=h.1idq7dh) Task Sheet (bảng kế hoạch công việc) 31

[5.3.](#_heading=h.42ddq1a) Checklist (bảng kiểm tra chức năng) 32

[5.3.1.](#_heading=h.2hio093) Kiểm tra dữ liệu nhập 32

[5.3.2.](#_heading=h.wnyagw) Kiểm tra các liên kết 33

**[TÀI LIỆU THAM KHẢO](#_heading=h.3gnlt4p)** [34](#_heading=h.3gnlt4p)

# **DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ**

1. **Thuật Ngữ và Từ Viết Tắt Trong Cơ Sở Dữ Liệu**

* **DBMS**: **Database Management System** – Hệ quản trị cơ sở dữ liệu.
* **RDBMS**: **Relational Database Management System** – Hệ quản trị cơ sở dữ liệu quan hệ.
* **SQL**: **Structured Query Language** – Ngôn ngữ truy vấn có cấu trúc.
* **NoSQL**: **Not Only SQL** – Cơ sở dữ liệu không phải quan hệ, ví dụ MongoDB.
* **ACID**: **Atomicity, Consistency, Isolation, Durability** – Các đặc tính của giao dịch cơ sở dữ liệu.
* **JOIN**: Lệnh trong SQL dùng để kết nối các bảng trong cơ sở dữ liệu.
* **Index**: Một cấu trúc dữ liệu trong cơ sở dữ liệu giúp tăng tốc độ truy vấn.
* **Normalization**: Quá trình tổ chức dữ liệu trong cơ sở dữ liệu để giảm sự dư thừa.

1. **Các Từ Viết Tắt và Thuật Ngữ Cơ Bản**

* MVC: Model-View-Controller – Kiến trúc phần mềm chia thành ba thành phần: Mô hình (Model), Giao diện người dùng (View), và Bộ điều khiển (Controller).
* Eloquent ORM: Object-Relational Mapping – Là một hệ thống ORM mạnh mẽ trong Laravel giúp bạn làm việc với cơ sở dữ liệu như là các đối tượng.
* CLI: Command Line Interface – Giao diện dòng lệnh. Laravel cung cấp một công cụ mạnh mẽ gọi là Artisan CLI để thực hiện các tác vụ như tạo controller, migration, và nhiều tác vụ khác.
* Artisan: Công cụ dòng lệnh của Laravel giúp tự động hóa các tác vụ lập trình thông qua các lệnh như php artisan make:model, php artisan migrate, v.v.
* Blade: Blade là templating engine của Laravel, giúp bạn xây dựng giao diện động một cách dễ dàng và nhanh chóng. Nó cung cấp các cú pháp đơn giản để làm việc với các dữ liệu động trong view.
* Env: Các tệp cấu hình trong Laravel thường được quản lý bằng cách sử dụng tệp .env để lưu trữ các biến môi trường (như thông tin cấu hình cơ sở dữ liệu, API keys, v.v.).
* Seeder: Là các tệp dùng để nhập dữ liệu mẫu vào cơ sở dữ liệu khi cần thiết (ví dụ: dữ liệu ban đầu cho các bảng trong cơ sở dữ liệu).
* Migration: Là công cụ để quản lý cấu trúc cơ sở dữ liệu trong Laravel. Migrations giúp bạn tạo và chỉnh sửa bảng cơ sở dữ liệu bằng cách sử dụng code thay vì làm thủ công.
* Middleware: Là các lớp trung gian giúp bạn kiểm tra và xử lý yêu cầu HTTP trước khi gửi đến Controller. Ví dụ: kiểm tra xác thực người dùng, kiểm tra quyền truy cập.
* Route: Là các định tuyến HTTP trong Laravel, xác định các URL và các hành động mà ứng dụng sẽ thực hiện khi người dùng truy cập các URL đó.
* Request: Là đối tượng chứa thông tin về yêu cầu HTTP của người dùng, như phương thức (GET, POST), dữ liệu gửi lên, v.v.
* Response: Là đối tượng chứa thông tin về phản hồi của ứng dụng khi người dùng yêu cầu một tài nguyên nào đó.
* Controller: Là lớp giúp xử lý các logic của ứng dụng khi nhận được yêu cầu từ các route. Controllers phân tách logic xử lý khỏi views.
* Service Provider: Là các lớp trong Laravel giúp đăng ký các dịch vụ hoặc các phụ thuộc của ứng dụng trong container (IoC container).
* Container (IoC): Inversion of Control – Mô hình quản lý các phụ thuộc (dependencies). Laravel sử dụng container để quản lý các lớp và đối tượng của ứng dụng.

1. **Các Tính Năng và Công Cụ Laravel**

* Queues: Hàng đợi giúp xử lý các tác vụ không đồng bộ như gửi email, xử lý ảnh, hoặc các tác vụ tốn thời gian khác mà không làm gián đoạn quá trình xử lý yêu cầu của người dùng.
* Events & Listeners: Laravel hỗ trợ mô hình Event-Listener, cho phép bạn xử lý các sự kiện và phản ứng lại với chúng, ví dụ như gửi thông báo khi có một người dùng mới đăng ký.
* Job: Tác vụ trong Laravel có thể được xử lý trong các hàng đợi (queue), giúp bạn tách biệt các tác vụ phức tạp hoặc lâu dài khỏi các thao tác yêu cầu chính.
* Testing: Laravel có hỗ trợ rất tốt cho việc Test ứng dụng. Laravel hỗ trợ các công cụ như PHPUnit để kiểm tra tính đúng đắn của ứng dụng và mã nguồn.
* Sanctum: Laravel Sanctum là một giải pháp nhẹ để bảo mật API trong các ứng dụng Laravel, cho phép cấp phát và xác thực các token API cho người dùng.
* Passport: Laravel Passport là một hệ thống hoàn chỉnh để xác thực OAuth2 cho API trong Laravel. Nó cung cấp khả năng tạo các token API cho người dùng và hệ thống của bạn.
* Validation: Là quy trình kiểm tra các dữ liệu đầu vào (request) của người dùng. Laravel cung cấp một hệ thống validation mạnh mẽ, dễ sử dụng để đảm bảo tính toàn vẹn của dữ liệu.
* Collection: Laravel cung cấp Collections, một bộ công cụ mạnh mẽ giúp bạn xử lý các mảng hoặc dữ liệu như một đối tượng, với các phương thức như map(), filter(), reduce(), v.v.
* Resource: Là một lớp giúp biến dữ liệu (thường là dữ liệu từ cơ sở dữ liệu) thành JSON hoặc các định dạng khác để trả về cho người dùng.

1. **Thuật Ngữ trong Laravel API**

* API Resource: Các đối tượng tài nguyên (Resource) trong Laravel giúp bạn định dạng dữ liệu JSON trả về từ API.
* Throttle: Throttle trong Laravel dùng để giới hạn số lần người dùng có thể gửi yêu cầu đến API trong một khoảng thời gian nhất định, nhằm bảo vệ API khỏi các cuộc tấn công DDOS hoặc sử dụng quá mức.
* CORS: Cross-Origin Resource Sharing – Chính sách bảo mật cho phép các tài nguyên trên web có thể được chia sẻ giữa các domain khác nhau. Laravel hỗ trợ cấu hình CORS dễ dàng.
* Rate Limiting: Giới hạn tần suất yêu cầu mà một người dùng có thể gửi đến API trong một khoảng thời gian nhất định.

1. **Các Từ Viết Tắt trong Laravel**

* DB: Database – Cơ sở dữ liệu. Laravel cung cấp một bộ công cụ giúp dễ dàng làm việc với cơ sở dữ liệu thông qua Eloquent ORM hoặc Query Builder.
* PDO: PHP Data Objects – Một cách để giao tiếp với cơ sở dữ liệu trong PHP, Laravel hỗ trợ PDO qua DB facade.
* JWT: JSON Web Token – Một chuẩn để truyền tải thông tin giữa các hệ thống qua các token, đặc biệt trong xác thực API.
* CSRF: Cross-Site Request Forgery – Tấn công mạo danh yêu cầu của người dùng từ một trang web khác. Laravel sử dụng token CSRF để bảo vệ ứng dụng.
* XSS: Cross-Site Scripting – Một loại tấn công bảo mật khi kẻ tấn công đưa mã JavaScript vào trang web của bạn. Laravel bảo vệ các form và đầu vào khỏi XSS.

1. **Thuật Ngữ Liên Quan đến Quản Lý Cơ Sở Dữ Liệu**

* Schema: Là một lớp trong Laravel để tạo hoặc thay đổi cấu trúc cơ sở dữ liệu. Nó hỗ trợ các phương thức như create(), table(), drop().
* Pivot Table: Là bảng liên kết giữa các bảng trong cơ sở dữ liệu khi bạn sử dụng mối quan hệ nhiều-nhiều (Many-to-Many).
* Seeder: Là các lớp được sử dụng để tạo dữ liệu mẫu vào cơ sở dữ liệu. Dữ liệu này có thể được sử dụng trong quá trình phát triển hoặc thử nghiệm.
* Factory: Các lớp giúp tạo dữ liệu mẫu (fake data) cho quá trình thử nghiệm hoặc phát triển.

1. **Thuật Ngữ và Tính Năng Mới**

* Laravel Mix: Một công cụ build dành cho các asset front-end (CSS, JavaScript, v.v.) tích hợp với Webpack. Laravel Mix giúp bạn dễ dàng biên dịch và tối ưu hóa các tài nguyên này.
* Livewire: Một thư viện của Laravel cho phép bạn xây dựng các ứng dụng web động, tương tác mà không cần phải viết nhiều JavaScript.
* Telescope: Là một công cụ giám sát của Laravel, cho phép bạn theo dõi các truy vấn cơ sở dữ liệu, các lỗi, các yêu cầu HTTP, và các sự kiện trong ứng dụng.
* Horizon: Là một công cụ giám sát các hàng đợi trong Laravel, giúp bạn dễ dàng theo dõi và quản lý các job trong queue.

1. **Các Loại Quan Hệ (Relationships)**

* One-to-One: Mối quan hệ một-một giữa hai bảng trong cơ sở dữ liệu (ví dụ: mỗi người dùng chỉ có một profile).
* One-to-Many: Mối quan hệ một-nhiều giữa hai bảng (ví dụ: một bài viết có nhiều bình luận).
* Many-to-Many: Mối quan hệ nhiều-nhiều giữa các bảng (ví dụ: một bài viết có thể có nhiều tag, và một tag có thể thuộc nhiều bài viết).
* Polymorphic Relationships: Mối quan hệ cho phép một mô hình có thể thuộc về nhiều mô hình khác nhau, như bình luận có thể thuộc về bài viết hoặc video.

1. **Các Công Cụ Kiểm Tra (Testing)**

* PHPUnit: Một thư viện PHP để viết các bài kiểm tra (unit test) cho mã nguồn. Laravel tích hợp sẵn PHPUnit để bạn kiểm tra tính đúng đắn của ứng dụng.
* Feature Test: Là bài kiểm tra kiểm tra các tính năng hoặc luồng ứng dụng (ví dụ: đăng nhập, thêm sản phẩm vào giỏ hàng, v.v.).
* Unit Test: Kiểm tra một đơn vị nhỏ của mã nguồn, chẳng hạn như một phương thức trong controller hoặc một hàm trong model.

1. **Các Tính Năng Khác**

* Job: Một lớp trong Laravel dùng để xử lý các tác vụ nặng hoặc các tác vụ không đồng bộ. Các job thường được đẩy vào queue và xử lý sau.
* Queue: Là hệ thống xử lý các tác vụ không đồng bộ, giúp ứng dụng không bị gián đoạn khi thực hiện các tác vụ lâu dài (ví dụ: gửi email, xử lý ảnh).
* Queue Worker: Là tiến trình giúp lấy các job từ hàng đợi và xử lý chúng.

1. **Các Thuật Ngữ Liên Quan đến Tối Ưu Hiệu Suất**

* Caching: Là việc lưu trữ tạm thời dữ liệu để tránh phải tính toán lại hoặc truy vấn cơ sở dữ liệu nhiều lần.
* Redis: Một hệ thống lưu trữ dữ liệu trong bộ nhớ (in-memory data store) thường được dùng trong Laravel để cache hoặc làm queue driver.
* Lazy Loading: Kỹ thuật trong Eloquent giúp giảm số lượng truy vấn đến cơ sở dữ liệu bằng cách chỉ tải các đối tượng liên quan khi cần thiết.
* Eager Loading: Kỹ thuật trong Eloquent giúp tải tất cả các đối tượng liên quan trong một truy vấn duy nhất, giảm thiểu số lượng truy vấn.

# **DANH MỤC BẢNG BIỂU**

# **DANH MỤC HÌNH ẢNH**

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

## Tổng quan về lập trình thiết kế giao diện website

## Giới thiệu về HTML5

HTML (HyperText Markup Language) là một ngôn ngữ đánh dấu văn bản được thiết kế ra để tạo các trang web với các đoạn thông tin được trình bày trên World Wide Web (www). HTML là một phần không thể thiếu của mạng Internet. HTML5 là phiên bản mới sửa đổi thứ 5 của HTML. Nó được coi là ngôn ngữ chuẩn mới nhất cho HTML, thay cho cả HTML4, XHTML và HTML DOM Level 2; được thiết kế đặc biệt để cung cấp nội dung phong phú mà không cần các plugin bổ sung. Nó cho phép 1 lớp ứng dụng web mới ra đời. Các phiên bản hiện tại hỗ trợ nội dung đa phương tiện và các chức năng offline mà không cần đến những công nghệ bản quyền đi kèm; cung cấp hầu như tất cả mọi thứ từ hình ảnh động, đồ họa, âm nhạc đến phim ảnh, và cũng có thể sử dụng để xây dựng các ứng dụng web phức tạp. Ngoài ra HTML5 cũng hộ trợ tích hợp website và các ứng dụng với PC, Smartphone….

## Giới thiệu về CSS3

CSS3 (Cascading Style Sheets Level 3) là phiên bản thứ ba của ngôn ngữ định kiểu được sử dụng để thiết kế giao diện và bố cục của các trang web. Đây là một phần quan trọng trong bộ ba công nghệ web cốt lõi, bao gồm HTML (xây dựng cấu trúc), CSS (thiết kế giao diện), và JavaScript (tương tác). CSS3 ra đời nhằm mở rộng và cải tiến các khả năng của CSS2, cung cấp nhiều tính năng mới, linh hoạt và hiệu quả hơn.

**Các tính năng nổi bật của CSS3**

1. **Phân chia thành các mô-đun**  
   CSS3 được tổ chức thành nhiều mô-đun, giúp việc phát triển và duy trì trở nên dễ dàng hơn. Một số mô-đun quan trọng bao gồm Selectors, Box Model, Backgrounds and Borders, và Text Effects.
2. **Hiệu ứng đồ họa**  
   CSS3 cho phép tạo ra các hiệu ứng bắt mắt như gradient, bóng mờ (box-shadow, text-shadow), và hiệu ứng chuyển động (transitions, animations) mà không cần sử dụng hình ảnh hoặc JavaScript.
3. **Hỗ trợ thiết kế phản hồi (Responsive Design)**  
   CSS3 giới thiệu Media Queries, giúp thiết kế giao diện web tương thích với nhiều thiết bị và kích thước màn hình khác nhau.
4. **Kiểu chữ nâng cao**  
   CSS3 hỗ trợ @font-face, cho phép sử dụng các font tùy chỉnh, mang lại sự đa dạng và sáng tạo trong thiết kế văn bản.
5. **Cải tiến bố cục**  
   CSS3 cung cấp các công cụ mạnh mẽ như Flexbox và Grid Layout, giúp dễ dàng tạo bố cục phức tạp mà vẫn giữ được tính linh hoạt.

**Ưu điểm của CSS3**

* **Giảm phụ thuộc vào JavaScript**: Với CSS3, nhiều hiệu ứng động có thể được thực hiện mà không cần viết mã JavaScript.
* **Tăng hiệu suất**: Giảm kích thước và số lượng tài nguyên cần tải, giúp trang web tải nhanh hơn.
* **Tính linh hoạt**: Cung cấp khả năng tùy biến giao diện và bố cục mạnh mẽ, phù hợp với mọi loại thiết bị và trình duyệt hiện đại.

**Ứng dụng thực tế**

CSS3 được sử dụng rộng rãi trong thiết kế giao diện người dùng, tạo hiệu ứng tương tác, và cải thiện trải nghiệm người dùng trên các nền tảng web. Với sự hỗ trợ ngày càng mạnh mẽ từ các trình duyệt, CSS3 là công cụ không thể thiếu trong ngành thiết kế web hiện nay.

Tóm lại, CSS3 không chỉ đơn thuần là một phiên bản nâng cấp của CSS2, mà còn là một bước tiến quan trọng, giúp các nhà phát triển web sáng tạo hơn và đáp ứng tốt hơn nhu cầu của người dùng trong kỷ nguyên số.

## Tổng quan về ngôn ngữ lập trình Javascript

JavaScript là một ngôn ngữ lập trình kịch bản (scripting language) phổ biến, được sử dụng rộng rãi để tạo các trang web động và tương tác. Được phát triển lần đầu tiên vào năm 1995 bởi Brendan Eich tại Netscape, JavaScript đã trở thành một phần quan trọng trong các công nghệ cốt lõi của web, cùng với HTML và CSS.

**Đặc điểm chính của JavaScript**

1. **Ngôn ngữ phía client và phía server**
   * **Phía client**: JavaScript chạy trên trình duyệt, cho phép tương tác trực tiếp với người dùng mà không cần tải lại trang.
   * **Phía server**: Với sự ra đời của Node.js, JavaScript đã mở rộng khả năng để xử lý logic phía server.
2. **Động và linh hoạt**  
   JavaScript là ngôn ngữ động, không cần khai báo kiểu dữ liệu, giúp viết mã nhanh chóng và dễ dàng hơn.
3. **Hỗ trợ lập trình hướng đối tượng, hàm và sự kiện**  
   JavaScript hỗ trợ nhiều kiểu lập trình, bao gồm lập trình hướng đối tượng (OOP), lập trình hàm (functional programming) và xử lý sự kiện (event-driven programming).
4. **Khả năng tích hợp cao**  
   JavaScript dễ dàng tích hợp với HTML và CSS để tạo các ứng dụng web phức tạp.

**Ưu điểm của JavaScript**

* **Tương tác người dùng tốt**: Cung cấp các tính năng như xác thực biểu mẫu, hiệu ứng động, và cập nhật nội dung mà không cần tải lại trang (AJAX).
* **Hiệu suất cao**: JavaScript được các trình duyệt hiện đại tối ưu hóa, giúp các ứng dụng chạy nhanh và mượt mà.
* **Đa nền tảng**: JavaScript có thể chạy trên mọi thiết bị và hệ điều hành thông qua trình duyệt hoặc môi trường Node.js.
* **Hệ sinh thái phong phú**: Với hàng triệu thư viện và framework như React, Angular, Vue.js, JavaScript mang lại sự linh hoạt và hiệu quả cho các nhà phát triển.

**Ứng dụng của JavaScript**

1. **Phát triển giao diện web**: Tạo các trang web động, tối ưu trải nghiệm người dùng.
2. **Phát triển ứng dụng di động**: Các framework như React Native cho phép xây dựng ứng dụng di động đa nền tảng.
3. **Ứng dụng server**: Với Node.js, JavaScript có thể xây dựng API và xử lý logic server.
4. **Phát triển trò chơi**: Một số công cụ như Phaser hoặc Babylon.js hỗ trợ tạo trò chơi trực tuyến.
5. **Ứng dụng IoT**: JavaScript cũng được sử dụng để lập trình các thiết bị IoT.

**Hạn chế của JavaScript**

* **Bảo mật**: Do chạy trực tiếp trên trình duyệt, mã JavaScript dễ bị khai thác nếu không được bảo vệ kỹ lưỡng.
* **Khác biệt giữa các trình duyệt**: Mặc dù tiêu chuẩn hóa ngày càng tốt, JavaScript vẫn có thể hoạt động khác nhau trên các trình duyệt cũ.

**Kết luận**

JavaScript là một công cụ không thể thiếu trong phát triển web hiện đại, mang lại sức mạnh và sự linh hoạt để tạo ra các ứng dụng và trang web phong phú, tương tác. Với sự phát triển không ngừng của công nghệ, vai trò của JavaScript ngày càng mở rộng, trở thành một trong những ngôn ngữ lập trình phổ biến và quan trọng nhất hiện nay.

## Thư viện Jquery

**Thư viện jQuery: Công cụ Hỗ Trợ Lập Trình JavaScript Hiệu Quả**

**Giới thiệu về jQuery**

jQuery là một thư viện JavaScript nhẹ, nhanh và mạnh mẽ, được phát triển bởi John Resig vào năm 2006. Mục tiêu chính của jQuery là đơn giản hóa việc thao tác với Document Object Model (DOM), xử lý sự kiện, thực hiện AJAX, và các tác vụ phổ biến khác trong JavaScript. Với cú pháp ngắn gọn và dễ hiểu, jQuery đã trở thành một công cụ phổ biến, giúp tăng tốc độ phát triển ứng dụng web.

**Các tính năng nổi bật của jQuery**

1. **Truy xuất và thao tác DOM dễ dàng**
   * jQuery cung cấp cú pháp ngắn gọn để truy cập và thao tác các phần tử HTML trong DOM.
   * Hỗ trợ các bộ chọn CSS giúp việc tìm kiếm phần tử nhanh chóng và trực quan.
2. **Xử lý sự kiện (Event Handling)**
   * Cung cấp các phương thức để gán, gỡ, và xử lý sự kiện một cách dễ dàng và nhất quán trên các trình duyệt.
   * Ví dụ: click(), hover(), focus() và on().
3. **Hiệu ứng và hoạt hình (Animations)**
   * jQuery cho phép tạo ra các hiệu ứng động như ẩn/hiện, trượt (slide), và mờ dần (fade) với cú pháp đơn giản.
   * Ví dụ: fadeIn(), slideUp(), animate().
4. **Hỗ trợ AJAX mạnh mẽ**
   * Dễ dàng thực hiện các yêu cầu AJAX để tải dữ liệu từ server mà không cần tải lại trang.
   * Ví dụ: $.ajax(), $.get(), $.post().
5. **Khả năng mở rộng**
   * jQuery hỗ trợ phát triển các plugin, giúp mở rộng tính năng và chia sẻ mã nguồn.
   * Có rất nhiều plugin miễn phí từ cộng đồng, hỗ trợ từ tạo slideshow đến xử lý biểu đồ.
6. **Khả năng tương thích trình duyệt**
   * jQuery giải quyết các vấn đề khác biệt về API và hành vi giữa các trình duyệt, giúp mã JavaScript hoạt động ổn định hơn.

**Ưu điểm của jQuery**

1. **Dễ học và sử dụng**: Cú pháp đơn giản, dễ tiếp cận với cả những người mới học lập trình web.
2. **Tiết kiệm thời gian**: Giảm bớt mã JavaScript cần viết, tăng hiệu quả lập trình.
3. **Khả năng tương thích cao**: Hỗ trợ đa trình duyệt, bao gồm cả các trình duyệt cũ.
4. **Hệ sinh thái phong phú**: Có nhiều plugin và cộng đồng hỗ trợ rộng lớn.

**Hạn chế của jQuery**

1. **Kích thước thư viện**: So với các framework hiện đại, jQuery có kích thước lớn hơn, không tối ưu cho các ứng dụng cần tốc độ tải nhanh.
2. **Không còn phù hợp cho các ứng dụng phức tạp**: Với sự xuất hiện của các framework hiện đại như React, Angular, và Vue.js, vai trò của jQuery đã giảm đi trong các dự án lớn.
3. **Học thêm JavaScript thuần**: Nhiều tính năng của jQuery hiện đã được hỗ trợ trong JavaScript thuần, giảm nhu cầu sử dụng thư viện.

**Ứng dụng của jQuery**

* Tạo hiệu ứng giao diện web (slider, lightbox, hiệu ứng hover).
* Xử lý AJAX để tải dữ liệu động từ server.
* Tương tác với biểu mẫu (form validation, autocomplete).
* Phát triển các tính năng giao diện người dùng tùy chỉnh.

**Kết luận**

jQuery từng là công cụ không thể thiếu trong lập trình web, giúp giảm bớt khó khăn khi thao tác DOM và xử lý sự kiện. Tuy nhiên, với sự phát triển của các framework JavaScript hiện đại và khả năng cải thiện của JavaScript thuần, vai trò của jQuery trong các dự án mới đã giảm đi. Dù vậy, nó vẫn là một công cụ hữu ích trong nhiều trường hợp và được duy trì rộng rãi trong các dự án truyền thống.

## Framework Bootstrap

**Framework Bootstrap: Công Cụ Hỗ Trợ Thiết Kế Web Hiện Đại**

**Giới thiệu về Bootstrap**

Bootstrap là một framework front-end phổ biến, được phát triển bởi Mark Otto và Jacob Thornton tại Twitter vào năm 2011. Framework này được thiết kế để đơn giản hóa và tăng tốc quá trình phát triển giao diện web. Bootstrap cung cấp sẵn các công cụ, thành phần, và kiểu dáng giúp lập trình viên dễ dàng xây dựng các trang web hiện đại, đáp ứng (responsive) và thẩm mỹ.

**Các đặc điểm chính của Bootstrap**

1. **Thiết kế đáp ứng (Responsive Design)**
   * Sử dụng hệ thống lưới (Grid System) để bố cục trang web tự động thích ứng với các kích thước màn hình khác nhau, từ máy tính đến điện thoại di động.
2. **Cung cấp các thành phần giao diện**
   * Bao gồm các thành phần như nút bấm (buttons), thanh điều hướng (navbar), biểu mẫu (forms), thẻ (cards), và nhiều yếu tố giao diện khác.
3. **Hỗ trợ CSS và JavaScript**
   * Bootstrap tích hợp sẵn các kiểu dáng CSS và các plugin JavaScript, giúp tạo hiệu ứng và xử lý tương tác dễ dàng.
4. **Dễ tùy chỉnh**
   * Cho phép chỉnh sửa và cấu hình thông qua các biến SASS/LESS hoặc file CSS riêng.
5. **Hỗ trợ các trình duyệt hiện đại**
   * Tương thích với hầu hết các trình duyệt phổ biến như Chrome, Firefox, Safari, và Microsoft Edge.

**Cấu trúc của Bootstrap**

1. **Hệ thống lưới (Grid System)**
   * Là nền tảng của Bootstrap, cho phép sắp xếp nội dung trong các hàng (rows) và cột (columns) linh hoạt.
2. **Thành phần CSS**
   * Bao gồm kiểu dáng cho typography, bảng (tables), biểu mẫu (forms), và nhiều yếu tố khác.
3. **Thành phần JavaScript**
   * Các plugin tích hợp như modal, dropdown, carousel, tooltip, giúp tăng tính năng động cho trang web.
4. **Tiện ích (Utilities)**
   * Cung cấp các lớp sẵn có để xử lý nhanh chóng các vấn đề về căn chỉnh, khoảng cách, hiển thị, và nhiều tính năng khác.

**Ưu điểm của Bootstrap**

1. **Tiết kiệm thời gian**
   * Với các thành phần và kiểu dáng dựng sẵn, lập trình viên không cần thiết kế từ đầu.
2. **Thân thiện với người mới học**
   * Bootstrap dễ học và sử dụng, với tài liệu chi tiết và cộng đồng lớn.
3. **Khả năng mở rộng**
   * Dễ dàng tích hợp với các công cụ khác hoặc tùy chỉnh theo nhu cầu.
4. **Đảm bảo tính nhất quán**
   * Giao diện trang web sẽ đồng nhất trên các trình duyệt và thiết bị.

**Hạn chế của Bootstrap**

1. **Tính đồng nhất cao**
   * Nhiều trang web sử dụng Bootstrap có thể trông giống nhau nếu không tùy chỉnh đủ nhiều.
2. **Kích thước lớn**
   * Gói Bootstrap đầy đủ có thể khá nặng, gây ảnh hưởng đến tốc độ tải trang.
3. **Học cách tùy chỉnh**
   * Nếu cần giao diện riêng biệt, việc tùy chỉnh Bootstrap có thể phức tạp.

**Ứng dụng của Bootstrap**

1. **Phát triển giao diện web nhanh chóng**: Sử dụng cho các dự án cần triển khai nhanh.
2. **Xây dựng giao diện đáp ứng**: Đảm bảo trang web hoạt động tốt trên mọi thiết bị.
3. **Phát triển prototype**: Bootstrap là lựa chọn tuyệt vời để tạo bản mẫu nhanh cho ứng dụng.

**Kết luận**

Bootstrap là một công cụ mạnh mẽ, giúp đơn giản hóa và tăng tốc quá trình phát triển giao diện web. Với sự hỗ trợ mạnh mẽ từ cộng đồng và khả năng mở rộng, Bootstrap vẫn là một trong những framework front-end phổ biến nhất hiện nay. Dù có một số hạn chế, nó vẫn phù hợp với hầu hết các dự án, từ cá nhân đến doanh nghiệp.

## Tổng quan về hệ quản trị cơ sở đữ liệu MySQL

## Giới thiệu về MySQL

**Giới thiệu về MySQL**

**Tổng quan**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở, được phát triển lần đầu vào năm 1995 bởi MySQL AB (sau này thuộc Oracle Corporation). Với khả năng lưu trữ, quản lý và truy xuất dữ liệu hiệu quả, MySQL là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp.

**Đặc điểm nổi bật của MySQL**

1. **Mã nguồn mở**
   * MySQL là phần mềm mã nguồn mở, có sẵn miễn phí. Tuy nhiên, cũng có các phiên bản thương mại với tính năng mở rộng dành cho doanh nghiệp.
2. **Hiệu suất cao và ổn định**
   * MySQL được thiết kế tối ưu hóa hiệu suất, giúp xử lý các khối lượng dữ liệu lớn với tốc độ nhanh.
3. **Hỗ trợ SQL**
   * Cung cấp cú pháp SQL chuẩn, giúp truy vấn và thao tác cơ sở dữ liệu dễ dàng.
4. **Tính di động**
   * MySQL hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux, và các hệ điều hành khác.
5. **Khả năng tích hợp mạnh mẽ**
   * MySQL dễ dàng tích hợp với nhiều ngôn ngữ lập trình như PHP, Python, Java, và .NET, làm cho nó trở thành lựa chọn phổ biến trong phát triển ứng dụng web.
6. **Bảo mật cao**
   * Hỗ trợ xác thực người dùng mạnh mẽ và quản lý quyền truy cập, giúp bảo vệ dữ liệu an toàn.

**Ứng dụng của MySQL**

1. **Phát triển ứng dụng web**
   * MySQL là sự lựa chọn phổ biến trong việc lưu trữ và quản lý dữ liệu của các trang web, đặc biệt khi kết hợp với PHP (LAMP stack).
2. **Quản lý dữ liệu doanh nghiệp**
   * Sử dụng trong hệ thống ERP, CRM hoặc các ứng dụng phân tích dữ liệu.
3. **Thương mại điện tử**
   * MySQL hỗ trợ các hệ thống giỏ hàng, quản lý sản phẩm và giao dịch trực tuyến.
4. **Lưu trữ nội dung đa phương tiện**
   * Sử dụng để lưu trữ và quản lý dữ liệu đa phương tiện như hình ảnh, video và âm thanh.

**Kết luận**

MySQL là một công cụ mạnh mẽ, linh hoạt và dễ sử dụng, phù hợp với nhiều loại ứng dụng từ nhỏ đến lớn. Với hiệu suất cao và khả năng hỗ trợ đa nền tảng, MySQL vẫn là một trong những hệ quản trị cơ sở dữ liệu được ưa chuộng nhất, đặc biệt trong các dự án phát triển web. Mặc dù có một số hạn chế, nó vẫn là lựa chọn tối ưu cho nhiều doanh nghiệp và nhà phát triển trên toàn thế giới.

## Ưu điểm của MySQL

1. **Dễ sử dụng**
   * Giao diện thân thiện và hỗ trợ tài liệu đầy đủ giúp người mới học dễ dàng tiếp cận.
2. **Cộng đồng lớn**
   * Với cộng đồng phát triển rộng khắp, người dùng có thể dễ dàng tìm kiếm sự hỗ trợ hoặc tài liệu.
3. **Hiệu năng tối ưu**
   * MySQL hoạt động tốt cho cả các ứng dụng nhỏ và các hệ thống lớn với lượng truy cập cao.
4. **Hỗ trợ đa ngôn ngữ**
   * Cung cấp khả năng xử lý dữ liệu trong nhiều ngôn ngữ khác nhau.

## Nhược điểm của MySQL

1. **Hạn chế trong các tác vụ phức tạp**
   * So với các hệ quản trị cơ sở dữ liệu khác như PostgreSQL, MySQL có thể thiếu tính năng trong một số trường hợp phức tạp.
2. **Giới hạn về khả năng mở rộng**
   * Mặc dù hiệu quả cho các ứng dụng tầm trung, MySQL có thể gặp khó khăn khi quản lý các hệ thống dữ liệu cực lớn.
3. **Phụ thuộc vào Oracle**
   * Sau khi thuộc sở hữu của Oracle, có một số lo ngại về tính bền vững và khả năng hỗ trợ lâu dài của MySQL.

## Tổng quan về ngôn ngữ lập trình phía máy chủ

## Giới thiệu về nền tảng công nghệ nguồn mở

**Giới thiệu về Nền Tảng Công Nghệ Nguồn Mở**

**Tổng quan về công nghệ nguồn mở**

Công nghệ nguồn mở (Open Source Technology) là các phần mềm, công cụ hoặc hệ thống mà mã nguồn của chúng được cung cấp công khai, cho phép người dùng tự do xem, sửa đổi, phân phối, và cải tiến theo nhu cầu. Đây là một mô hình phát triển phần mềm phổ biến, được xây dựng dựa trên tinh thần chia sẻ và hợp tác trong cộng đồng.

Các nền tảng công nghệ nguồn mở đóng vai trò quan trọng trong ngành công nghệ thông tin, tạo điều kiện phát triển nhanh chóng các ứng dụng và giải pháp, đồng thời giúp giảm chi phí và tăng cường sự minh bạch.

**Đặc điểm chính của nền tảng nguồn mở**

1. **Mã nguồn công khai**
   * Mã nguồn được cung cấp miễn phí và công khai, cho phép người dùng tải xuống, chỉnh sửa và sử dụng.
2. **Cộng đồng phát triển mạnh mẽ**
   * Phần lớn các dự án nguồn mở được duy trì và cải tiến bởi một cộng đồng các nhà phát triển trên toàn cầu.
3. **Minh bạch và bảo mật**
   * Người dùng có thể kiểm tra mã nguồn để phát hiện các lỗ hổng bảo mật hoặc cải thiện hiệu năng.
4. **Khả năng tùy chỉnh cao**
   * Người dùng có thể thay đổi mã nguồn để đáp ứng các yêu cầu riêng biệt.

**Ưu điểm của công nghệ nguồn mở**

1. **Chi phí thấp**
   * Hầu hết các phần mềm nguồn mở đều miễn phí, giúp giảm đáng kể chi phí phát triển và triển khai.
2. **Tự do và linh hoạt**
   * Người dùng có thể tùy chỉnh phần mềm theo nhu cầu mà không phụ thuộc vào nhà cung cấp.
3. **Cải tiến liên tục**
   * Với sự tham gia của cộng đồng lớn, các nền tảng nguồn mở thường được cập nhật và cải tiến nhanh chóng.
4. **Tính bền vững**
   * Vì mã nguồn mở nên các dự án không bị phụ thuộc vào một tổ chức duy nhất; người dùng có thể tiếp tục phát triển ngay cả khi dự án gốc ngừng hoạt động.
5. **Cộng đồng hỗ trợ**
   * Cộng đồng nguồn mở thường cung cấp tài liệu chi tiết và hỗ trợ qua các diễn đàn, nhóm trực tuyến.

**Hạn chế của công nghệ nguồn mở**

1. **Yêu cầu kỹ năng kỹ thuật cao**
   * Việc triển khai và tùy chỉnh các phần mềm nguồn mở thường đòi hỏi người dùng có kiến thức chuyên môn.
2. **Không có hỗ trợ chính thức**
   * Mặc dù cộng đồng hỗ trợ rất tốt, một số dự án không cung cấp dịch vụ hỗ trợ chuyên nghiệp.
3. **Khả năng tương thích**
   * Trong một số trường hợp, phần mềm nguồn mở có thể không tương thích tốt với các hệ thống hoặc công cụ độc quyền.

**Ứng dụng của công nghệ nguồn mở**

1. **Hệ điều hành**
   * Linux (Ubuntu, CentOS, Debian) là hệ điều hành nguồn mở phổ biến được sử dụng trong các máy chủ, máy tính cá nhân và thiết bị IoT.
2. **Hệ quản trị cơ sở dữ liệu**
   * MySQL, PostgreSQL, MongoDB là các hệ quản trị cơ sở dữ liệu mạnh mẽ và phổ biến.
3. **Framework và thư viện**
   * Django, Laravel, Spring, React, và Angular là các framework phát triển web và ứng dụng nguồn mở nổi tiếng.
4. **Phần mềm ứng dụng**
   * LibreOffice, GIMP, và VLC Media Player là các ví dụ về phần mềm ứng dụng nguồn mở.
5. **Công cụ phát triển và DevOps**
   * Docker, Kubernetes, Jenkins, và Git là các công cụ hỗ trợ phát triển phần mềm và quản lý hệ thống.

**Kết luận**

Nền tảng công nghệ nguồn mở đã và đang đóng góp quan trọng vào sự phát triển của ngành công nghệ thông tin. Với khả năng tùy chỉnh cao, chi phí thấp, và sự hỗ trợ từ cộng đồng, các giải pháp nguồn mở là lựa chọn lý tưởng cho cá nhân, tổ chức, và doanh nghiệp. Tuy nhiên, việc sử dụng công nghệ nguồn mở cũng đòi hỏi kỹ năng và kiến thức để khai thác hiệu quả tiềm năng của nó.

## Giới thiệu về ngôn ngữ lập trình PHP

**Giới thiệu về ngôn ngữ lập trình PHP**

**Tổng quan về PHP**

PHP (viết tắt của "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình kịch bản phía máy chủ (server-side scripting language) được thiết kế chủ yếu để phát triển web. PHP được tạo ra vào năm 1994 bởi Rasmus Lerdorf và nhanh chóng trở thành một trong những ngôn ngữ lập trình phổ biến nhất trong lĩnh vực phát triển ứng dụng web.

PHP được tối ưu hóa để hoạt động với HTML và hỗ trợ nhiều hệ quản trị cơ sở dữ liệu, đặc biệt là MySQL, làm cho nó trở thành công cụ mạnh mẽ trong việc xây dựng các trang web động và ứng dụng web.

**Đặc điểm nổi bật của PHP**

1. **Mã nguồn mở và miễn phí**
   * PHP là một ngôn ngữ mã nguồn mở, miễn phí sử dụng và dễ dàng tiếp cận.
2. **Tích hợp tốt với HTML**
   * PHP được thiết kế để nhúng trực tiếp vào mã HTML, giúp đơn giản hóa việc tạo ra các trang web động.
3. **Hoạt động trên nhiều nền tảng**
   * PHP chạy tốt trên các hệ điều hành như Windows, macOS, Linux và nhiều nền tảng khác.
4. **Hỗ trợ nhiều hệ quản trị cơ sở dữ liệu**
   * PHP tích hợp dễ dàng với MySQL, PostgreSQL, SQLite, Oracle và nhiều hệ cơ sở dữ liệu khác.
5. **Thư viện phong phú và cộng đồng lớn**
   * PHP có nhiều thư viện sẵn có, cộng đồng phát triển rộng lớn, hỗ trợ mạnh mẽ qua các diễn đàn và tài liệu trực tuyến.

**Ưu điểm của PHP**

1. **Dễ học và sử dụng**
   * Cú pháp đơn giản và tài liệu phong phú giúp người mới học dễ dàng tiếp cận.
2. **Hiệu suất cao**
   * PHP được tối ưu hóa cho các ứng dụng web, cho phép xử lý hàng triệu yêu cầu mỗi ngày trên các trang web lớn.
3. **Khả năng mở rộng**
   * PHP hỗ trợ nhiều framework như Laravel, Symfony, CodeIgniter, giúp tăng tốc độ phát triển và tổ chức mã nguồn tốt hơn.
4. **Chi phí thấp**
   * Do mã nguồn mở và miễn phí, PHP giúp giảm chi phí phát triển cho các doanh nghiệp và cá nhân.
5. **Tích hợp mạnh mẽ với các công nghệ web**
   * PHP hoạt động tốt với các giao thức HTTP, FTP, và các công nghệ khác như XML, JSON.

**Hạn chế của PHP**

1. **Hiệu năng không cao trong các ứng dụng lớn**
   * Mặc dù hiệu quả với các trang web vừa và nhỏ, PHP có thể gặp khó khăn khi xử lý các ứng dụng phức tạp hoặc yêu cầu hiệu năng cao.
2. **Bảo mật cần được cải thiện**
   * Do tính phổ biến, PHP là mục tiêu của nhiều cuộc tấn công, đòi hỏi lập trình viên phải áp dụng các biện pháp bảo mật nghiêm ngặt.
3. **Cú pháp không nhất quán**
   * Một số nhà phát triển chỉ trích sự thiếu nhất quán trong cách đặt tên hàm và cú pháp của PHP.

**Ứng dụng của PHP**

1. **Phát triển trang web động**
   * PHP thường được sử dụng để xây dựng các trang web động, như blog, cổng thông tin, và diễn đàn.
2. **Hệ thống quản lý nội dung (CMS)**
   * Các CMS nổi tiếng như WordPress, Joomla, Drupal đều được xây dựng trên PHP.
3. **Thương mại điện tử**
   * PHP hỗ trợ các nền tảng như Magento, WooCommerce, giúp xây dựng các trang web thương mại điện tử.
4. **Phát triển API**
   * PHP hỗ trợ tạo và quản lý các API để giao tiếp giữa các ứng dụng.
5. **Ứng dụng web phức tạp**
   * Với sự hỗ trợ từ các framework, PHP được sử dụng để phát triển các ứng dụng web lớn, như mạng xã hội hoặc hệ thống đặt vé.

**Kết luận**

PHP là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và phù hợp với nhiều loại dự án, từ trang web đơn giản đến ứng dụng web phức tạp. Với khả năng tích hợp tốt, cộng đồng lớn và tài liệu phong phú, PHP vẫn là lựa chọn hàng đầu của nhiều nhà phát triển web trên toàn thế giới. Dù gặp một số hạn chế, PHP vẫn giữ vị trí quan trọng trong lĩnh vực phát triển web hiện đại.

## Tổng quan về Framework PHP Laravel

**Tổng quan về Framework PHP Laravel**

**Giới thiệu Laravel**

Laravel là một framework PHP mã nguồn mở, được phát triển bởi Taylor Otwell vào năm 2011. Laravel được thiết kế để hỗ trợ phát triển các ứng dụng web dễ dàng, nhanh chóng và có cấu trúc rõ ràng. Dựa trên mô hình MVC (Model-View-Controller), Laravel cung cấp một hệ sinh thái mạnh mẽ với các công cụ và tính năng phong phú để tăng năng suất và tính tổ chức trong phát triển ứng dụng.

**Đặc điểm nổi bật của Laravel**

1. **Mô hình MVC**
   * Laravel sử dụng mô hình MVC giúp tổ chức mã nguồn rõ ràng, dễ bảo trì và mở rộng.
2. **Blade Template Engine**
   * Laravel tích hợp Blade, một công cụ tạo giao diện mạnh mẽ, cho phép tạo các giao diện linh hoạt và dễ quản lý.
3. **Hệ thống định tuyến mạnh mẽ**
   * Laravel cung cấp các tính năng định tuyến đơn giản và mạnh mẽ, giúp quản lý URL và phản hồi dễ dàng.
4. **Eloquent ORM (Object-Relational Mapping)**
   * Laravel hỗ trợ Eloquent, một công cụ ORM giúp làm việc với cơ sở dữ liệu trực quan, giảm thiểu việc viết SQL thủ công.
5. **Hệ thống Migration và Seeding**
   * Laravel cung cấp các công cụ migration và seeding để quản lý và khởi tạo cơ sở dữ liệu một cách hiệu quả.
6. **Bảo mật tích hợp**
   * Laravel hỗ trợ các tính năng bảo mật như mã hóa, xác thực người dùng, chống CSRF, và quản lý quyền truy cập.
7. **Công cụ Artisan**
   * Artisan là công cụ dòng lệnh của Laravel, giúp tự động hóa các tác vụ phổ biến như tạo controller, migration, và quản lý cache.
8. **Hỗ trợ API và RESTful**
   * Laravel cung cấp các tính năng tích hợp để phát triển API một cách nhanh chóng, bao gồm định tuyến API, xác thực và xử lý dữ liệu JSON.

**Ưu điểm của Laravel**

1. **Phát triển nhanh chóng**
   * Với các công cụ và tính năng dựng sẵn, Laravel tăng tốc độ phát triển ứng dụng.
2. **Cộng đồng lớn và tài liệu phong phú**
   * Laravel có một cộng đồng phát triển rộng lớn, cung cấp sự hỗ trợ mạnh mẽ thông qua tài liệu, diễn đàn và các khóa học trực tuyến.
3. **Tích hợp dễ dàng với các công nghệ khác**
   * Laravel tích hợp tốt với các công cụ front-end như Vue.js, React, và Angular, cũng như các hệ thống cơ sở dữ liệu.
4. **Khả năng mở rộng cao**
   * Laravel hỗ trợ xây dựng các ứng dụng từ nhỏ đến lớn, với khả năng mở rộng và tùy chỉnh dễ dàng.
5. **Tính bảo mật cao**
   * Laravel cung cấp các biện pháp bảo mật mạnh mẽ để bảo vệ ứng dụng khỏi các cuộc tấn công phổ biến.

**Hạn chế của Laravel**

1. **Hiệu năng thấp hơn so với các framework khác**
   * Laravel có thể chậm hơn so với một số framework nhẹ khác khi xử lý các ứng dụng đòi hỏi hiệu năng cao.
2. **Đường cong học tập**
   * Mặc dù tài liệu phong phú, người mới học có thể cần thời gian để làm quen với hệ sinh thái phức tạp của Laravel.
3. **Yêu cầu môi trường mạnh mẽ hơn**
   * Các ứng dụng Laravel có thể yêu cầu tài nguyên máy chủ lớn hơn, đặc biệt đối với các ứng dụng phức tạp.

**Ứng dụng của Laravel**

1. **Trang web thương mại điện tử**
   * Laravel được sử dụng để xây dựng các hệ thống quản lý cửa hàng trực tuyến với các tính năng giỏ hàng, thanh toán và quản lý sản phẩm.
2. **Hệ thống quản lý nội dung (CMS)**
   * Laravel có thể được sử dụng để phát triển các CMS tùy chỉnh.
3. **Ứng dụng quản lý doanh nghiệp**
   * Laravel được sử dụng trong các hệ thống quản lý dự án, ERP, và CRM.
4. **Phát triển API**
   * Laravel là một trong những framework tốt nhất để xây dựng các API mạnh mẽ và linh hoạt.
5. **Ứng dụng web thời gian thực**
   * Laravel hỗ trợ xây dựng các ứng dụng với tính năng thời gian thực, như trò chuyện hoặc thông báo trực tuyến.

**Kết luận**

Laravel là một framework PHP mạnh mẽ, phù hợp cho cả người mới học và các nhà phát triển chuyên nghiệp. Với hệ sinh thái đa dạng, cộng đồng hỗ trợ tốt và tính năng phong phú, Laravel không chỉ giúp tăng tốc phát triển ứng dụng mà còn đảm bảo ứng dụng có chất lượng cao, dễ bảo trì và mở rộng. Đây là một trong những framework PHP hàng đầu được sử dụng rộng rãi trong phát triển web hiện nay.

# **CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI**

## Đặt vấn đề

 **Hạn chế phạm vi tiếp cận:** Khách hàng tiềm năng bị giới hạn trong khu vực địa phương.

 **Cạnh tranh gay gắt:** Khó cạnh tranh với các doanh nghiệp lớn đã có nền tảng online mạnh mẽ.

 **Khó khăn trong quản lý:** Không có hệ thống quản lý sản phẩm, đơn hàng, hoặc khách hàng hiệu quả.

 **Thói quen tiêu dùng:** Nhiều khách hàng ưa chuộng mua sắm trực tuyến hơn mua sắm trực tiếp.

## Hệ thống hiện tại

**1. Frontend (Giao diện người dùng):**

* **Giao diện người dùng (UI):**
  + Trang chủ, danh mục sản phẩm, chi tiết sản phẩm.
  + Giỏ hàng, thanh toán, và các chức năng tìm kiếm.
  + Trang liên hệ, đăng nhập/đăng ký, và thông tin tài khoản.
  + Giao diện thân thiện với người dùng, responsive (tương thích với mọi thiết bị).
* **Công nghệ frontend phổ biến:**
  + **HTML, CSS, JavaScript:** Cơ bản để tạo giao diện và sự tương tác.
  + **React.js / Vue.js / Angular:** Các framework và thư viện giúp xây dựng giao diện động và cải thiện trải nghiệm người dùng.

**2. Backend (Máy chủ và quản lý hệ thống):**

* **Máy chủ (Server):**
  + **Web Server:** Nơi xử lý các yêu cầu HTTP từ trình duyệt người dùng. Ví dụ: Apache, Nginx.
  + **Ứng dụng Web (Backend):** Xử lý các yêu cầu từ người dùng như xử lý giỏ hàng, thanh toán, và quản lý đơn hàng.
  + **Cơ sở dữ liệu:** Lưu trữ thông tin sản phẩm, đơn hàng, khách hàng, và dữ liệu liên quan.
* **Công nghệ backend phổ biến:**
  + **Node.js (với Express.js):** Hệ thống server-side mạnh mẽ và nhanh chóng.
  + **PHP (với Laravel, WordPress):** Dễ sử dụng cho các website nhỏ với các CMS và framework phổ biến.
  + **Ruby on Rails:** Framework giúp phát triển nhanh các ứng dụng web.

**3. Cơ sở dữ liệu (Database):**

* **Cơ sở dữ liệu quan hệ (RDBMS):**
  + **MySQL, PostgreSQL:** Các hệ quản trị cơ sở dữ liệu phổ biến cho ứng dụng bán hàng nhỏ, lưu trữ dữ liệu sản phẩm, đơn hàng, và thông tin khách hàng.
* **Cơ sở dữ liệu NoSQL (được sử dụng trong các trường hợp đặc biệt):**
  + **MongoDB:** Dễ dàng mở rộng và phù hợp với dữ liệu phi cấu trúc hoặc không quan hệ.

**4. Tính năng chính của hệ thống hiện tại:**

* **Quản lý sản phẩm:**
  + Thêm, sửa, xóa sản phẩm từ bảng quản trị.
  + Các đặc tính của sản phẩm như tên, mô tả, giá cả, hình ảnh, và tồn kho.
* **Quản lý giỏ hàng và thanh toán:**
  + Cho phép khách hàng thêm sản phẩm vào giỏ hàng, thay đổi số lượng, và thanh toán online qua các cổng thanh toán như PayPal, Stripe, hoặc thanh toán COD (trả tiền khi nhận hàng).
* **Quản lý đơn hàng:**
  + Quản lý các đơn hàng từ khi khách hàng đặt cho đến khi giao hàng.
  + Thông báo trạng thái đơn hàng (đang xử lý, đã giao, bị hủy).
* **Quản lý khách hàng:**
  + Cho phép khách hàng đăng ký, đăng nhập và quản lý tài khoản của mình.
  + Theo dõi lịch sử mua hàng và thông tin giao hàng.

**5. Các công nghệ liên quan (Tùy chọn):**

* **Công cụ phân tích và theo dõi:**
  + **Google Analytics:** Theo dõi hành vi người dùng và hiệu suất trang web.
  + **Hotjar / Crazy Egg:** Phân tích hành vi người dùng qua bản đồ nhiệt và các cuộc khảo sát.
* **Email Marketing:**
  + **Mailchimp, SendGrid:** Gửi email quảng cáo, thông báo đơn hàng, và khuyến mãi.
* **SEO (Search Engine Optimization):**
  + Tối ưu hóa website để nâng cao vị trí trên công cụ tìm kiếm, giúp khách hàng dễ dàng tìm thấy sản phẩm của bạn.

**6. Hosting và Domain:**

* **Hosting:**
  + Lựa chọn một nhà cung cấp dịch vụ hosting như **Bluehost, HostGator, DigitalOcean, AWS** tùy theo nhu cầu về hiệu suất và khả năng mở rộng.
* **Tên miền (Domain):**
  + Mua và đăng ký tên miền cho website của bạn, ví dụ: **[www.tenmiencuaban.com](http://www.tenmiencuaban.com" \t "_new)**.

**Tóm tắt hệ thống hiện tại cho website bán hàng nhỏ:**

* **Frontend:** HTML, CSS, JavaScript (React.js, Vue.js, hoặc Angular).
* **Backend:** Node.js (Express), PHP (Laravel), Ruby on Rails.
* **Cơ sở dữ liệu:** MySQL hoặc PostgreSQL.
* **Tính năng:** Quản lý sản phẩm, giỏ hàng, thanh toán, đơn hàng, và khách hàng.
* **Hosting và Domain:** Các dịch vụ lưu trữ như DigitalOcean, AWS, Bluehost.
* **Công cụ bổ trợ:** Google Analytics, email marketing, SEO.

## Hệ thống đề xuất

**1. Frontend (Giao diện người dùng)**

**Công nghệ đề xuất:**

* **HTML5, CSS3, JavaScript**: Các ngôn ngữ cơ bản để xây dựng giao diện.
* **Frameworks/Thư viện:**
  + **React.js**: Tạo giao diện người dùng linh hoạt, có thể tái sử dụng các thành phần.
  + **Vue.js**: Dễ sử dụng và tối ưu cho các ứng dụng nhỏ và vừa.
  + **Tailwind CSS**: Framework CSS giúp dễ dàng thiết kế giao diện đẹp mà không cần viết quá nhiều CSS.
  + **Bootstrap**: Framework CSS giúp xây dựng giao diện nhanh chóng, dễ dàng responsive.

**Tính năng đề xuất:**

* **Trang chủ**: Giới thiệu về cửa hàng và các sản phẩm nổi bật.
* **Danh mục sản phẩm**: Hiển thị các sản phẩm theo từng loại, có bộ lọc theo giá, đánh giá, v.v.
* **Chi tiết sản phẩm**: Mô tả chi tiết, hình ảnh sản phẩm, đánh giá từ khách hàng, và các sản phẩm liên quan.
* **Giỏ hàng**: Quản lý sản phẩm trong giỏ, thay đổi số lượng, xóa sản phẩm.
* **Thanh toán**: Cổng thanh toán qua PayPal, Stripe, hoặc COD (thanh toán khi nhận hàng).
* **Tìm kiếm sản phẩm**: Hỗ trợ tìm kiếm sản phẩm dễ dàng qua tên, giá hoặc thể loại.
* **Đăng ký, đăng nhập và quản lý tài khoản**: Người dùng có thể tạo tài khoản và quản lý đơn hàng của mình.

**2. Backend (Máy chủ và quản lý hệ thống)**

**Công nghệ đề xuất:**

* **Node.js + Express.js**:
  + **Node.js** giúp xử lý các yêu cầu bất đồng bộ nhanh chóng, rất phù hợp với ứng dụng bán hàng nhỏ và yêu cầu hiệu suất cao.
  + **Express.js** là một framework nhẹ, nhanh chóng và dễ dàng triển khai các API RESTful.
* **Hoặc PHP + Laravel**:
  + **PHP** là ngôn ngữ phổ biến và dễ triển khai cho các website bán hàng nhỏ.
  + **Laravel** là framework PHP mạnh mẽ, dễ sử dụng và cung cấp nhiều tính năng hỗ trợ quản lý sản phẩm, đơn hàng, khách hàng, và bảo mật.
* **Cơ sở dữ liệu đề xuất:**
  + **MySQL** hoặc **PostgreSQL**: Đây là các cơ sở dữ liệu quan hệ phổ biến, dễ triển khai cho các website bán hàng nhỏ.
  + **MongoDB** (nếu cần xử lý dữ liệu phi cấu trúc): Dành cho các website có nhu cầu lưu trữ dữ liệu linh hoạt, không phụ thuộc vào các bảng quan hệ.

**Tính năng đề xuất:**

* **Quản lý sản phẩm**: Thêm, sửa, xóa, và phân loại sản phẩm.
* **Quản lý đơn hàng**: Theo dõi trạng thái đơn hàng từ khi đặt cho đến khi giao hàng.
* **Quản lý khách hàng**: Lưu trữ thông tin khách hàng, lịch sử mua hàng, và thông tin liên lạc.
* **Quản lý kho**: Cập nhật số lượng sản phẩm còn trong kho và thông báo khi hết hàng.

**3. Cổng thanh toán**

**Cổng thanh toán đề xuất:**

* **PayPal**: Đơn giản, phổ biến, và an toàn.
* **Stripe**: Hỗ trợ nhiều loại thẻ tín dụng và dễ tích hợp vào website.
* **COD (Thanh toán khi nhận hàng)**: Cung cấp thêm lựa chọn thanh toán cho khách hàng.

**4. Hosting và Domain**

**Đề xuất dịch vụ hosting:**

* **DigitalOcean / AWS**: Cung cấp dịch vụ cloud hosting mạnh mẽ, có thể mở rộng khi cần.
* **Bluehost / Hostgator**: Các dịch vụ hosting phổ biến, dễ sử dụng, phù hợp với website nhỏ và vừa.
* **VPS Hosting**: Nếu có ngân sách và yêu cầu cao về hiệu suất và bảo mật.

**Tên miền (Domain):**

* **Tên miền ngắn gọn, dễ nhớ**: Ví dụ: **[www.tenmiencuaban.com](http://www.tenmiencuaban.com" \t "_new)**. Nên chọn tên miền với đuôi **.com**, **.vn**, hoặc **.store** để phù hợp với mô hình bán hàng.

**5. SEO và Marketing**

**Công cụ SEO đề xuất:**

* **Yoast SEO (cho WordPress)**: Giúp tối ưu hóa website của bạn cho công cụ tìm kiếm.
* **Google Search Console & Google Analytics**: Theo dõi hiệu suất và hành vi người dùng, cũng như tối ưu hóa SEO cho website.

**Email Marketing:**

* **Mailchimp**: Tích hợp dễ dàng với các website và giúp gửi email khuyến mãi, thông báo đơn hàng cho khách hàng.
* **SendGrid**: Một lựa chọn khác để gửi email giao dịch và marketing.

**6. Bảo mật**

**Đề xuất bảo mật cho website:**

* **SSL (HTTPS)**: Cung cấp chứng chỉ SSL cho website để mã hóa dữ liệu truyền tải và bảo mật thông tin khách hàng.
* **Xác thực người dùng**: Sử dụng các biện pháp bảo mật như mã hóa mật khẩu và xác thực hai yếu tố (2FA) cho tài khoản người dùng.
* **Firewall & Antivirus**: Đảm bảo hệ thống của bạn được bảo vệ khỏi các tấn công từ bên ngoài.

**7. Tính năng mở rộng trong tương lai**

* **Quản lý quảng cáo:** Tính năng chạy quảng cáo trong website, hiển thị các sản phẩm quảng cáo cho khách hàng.
* **Chatbot hỗ trợ khách hàng**: Cung cấp tính năng hỗ trợ trực tuyến qua chatbot để trả lời nhanh các câu hỏi thường gặp.

**Tóm tắt hệ thống đề xuất:**

* **Frontend:** React.js, Vue.js, hoặc Angular + Tailwind CSS.
* **Backend:** Node.js (Express.js) hoặc PHP (Laravel).
* **Cơ sở dữ liệu:** MySQL hoặc PostgreSQL.
* **Thanh toán:** PayPal, Stripe, và COD.
* **Hosting:** DigitalOcean, AWS, hoặc VPS Hosting.
* **SEO & Marketing:** Google Analytics, Yoast SEO, Mailchimp.
* **Bảo mật:** SSL, xác thực người dùng, firewall.

## Giới hạn của hệ thống

**Tài nguyên hệ thống hạn chế:** Dung lượng lưu trữ, băng thông và tài nguyên máy chủ có thể bị giới hạn.

**Tính năng đơn giản:** Chỉ hỗ trợ các tính năng cơ bản cho cửa hàng nhỏ.

**Khả năng mở rộng hạn chế:** Khó mở rộng khi lượng khách hàng và sản phẩm tăng lên.

**Bảo mật cơ bản:** Thiếu các biện pháp bảo mật nâng cao và phân quyền chi tiết.

**Trải nghiệm người dùng không tối ưu:** Tốc độ tải trang và khả năng tương thích thiết bị di động có thể không tốt.

**Phân tích dữ liệu hạn chế:** Thiếu khả năng phân tích và báo cáo tự động nâng cao.

## Yêu cầu về phần cứng và phần mềm

## Yêu cầu tối thiểu

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram: 4GB  \* HDD >=500MB free  \* CPU: Intel Core i3 hoặc AMD Ryzen 3 (dual-core).  \* Connect the Internet. | \* Ram: 4GB  \* CPU : Intel Core i3  \* Connect the Internet. |
| Software | \* Hệ điều hành:  **Linux:** Ubuntu Server, CentOS, Debian, hoặc Red Hat Enterprise Linux (RHEL).   * Ưu tiên Linux vì hiệu suất cao, bảo mật tốt, và miễn phí.   **Windows Server:** Phù hợp nếu ứng dụng cần tích hợp với các dịch vụ của Microsoft (IIS, .NET Framework).  \* SQL Server: SQL Server Express Edition  \* NetFramework: Ver 4.8 | \* Hệ điều hành:  **Máy tính:** Windows 10/11, macOS 10.12+, Linux (Ubuntu, Fedora).  **Thiết bị di động:** Android 8.0+ hoặc iOS 12.0+  \* IE 8.0, Firefox 4.0, Chrome 15….. |

## Yêu cầu đề nghị

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram: >= 2GB  \* HDD: >= 1GB free  \* CPU: Intel Core 2 Duo  \* Connect the Internet. | \* Ram: >=2GB  \* CPU: 0 yêu cầu  \*Connect the Internet. |
| Software | \* Hệ điều hành: Windows Server Windows Server 2019 hoặc 2022 Standard Edition.  \* SQL Server :**SQL Server Express Edition** hoặc **SQL Server Web Edition**  \* Net Framework: .NET Framework 4.8.1 | \* Từ Windows XP / Vista / 7 / 8 / 8.1… 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**

## Người dùng của hệ thống

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

* Guest (Khách vãng lai)
* Member (Thành viên)
* Admin (Quản trị hệ thống).

## Chức năng của hệ thống

* + 1. ***Guest*** *(Chức năng dành cho khách vãng lai)*
* **Xem sản phẩm**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể xem thông tin chi tiết từng loại giày như tên giày, kích cỡ, màu sắc, giá cả… |
| **Nhập** | Lựa chọn loại giày cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* **Duyệt Sản Phẩm**
* **Thêm Sản Phẩm vào Giỏ Hàng**
* **Thanh Toán**
* **Quản Lý Đơn Hàng Cơ Bản**
* **Được Tham Gia Chương Trình Khuyến Mại**
* **Đánh Giá Sản Phẩm (Có thể bị giới hạn)**
* **Đăng ký thành viên**

|  |  |
| --- | --- |
| **Mô tả** | Khách vãng lai có thể đăng ký làm thành viên. |
| **Nhập** | Đưa vào các thông tin đã nhập trong form để kiểm tra. |
| **Xử lý** | * 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. ***Member functions*** *(Chức năng dành cho thành viên)*

Thành viên chính thức của website sẽ có các chức năng giống như khách vãng lai như:

* + Xem sản phẩm.
  + Tìm kiếm sản phẩm.
  + Lựa chọn sản phẩm vào giỏ hàng.

***Ngoài ra thành viên chính thức còn có các chức năng sau***:

* **Đăng nhập**

|  |  |
| --- | --- |
| **Mô tả** | Đăng nhập vào website bằng cách nhập username và password đã đăng ký. |
| **Nhập** | Nhập username và mật khẩu đã đăng ký. |
| **Xử lý** | Truy cập kiểm tra xem tài khoản có tồn tại hay không. |
| **Xuất** | Hiển thị thông báo đăng nhập thành công khi khách hàng nhập đúng thông tin đã đăng ký và quay về trang chủ. Thông báo sai thông tin đăng nhập nếu tài khoản chưa đăng ký hoặc bị khóa, yêu cầu nhập lại thông tin đăng nhập! |

* **Đăng xuất**

|  |  |
| --- | --- |
| **Mô tả** | Đăng xuất khỏi website. |
| **Nhập** | Nhấn nút thoát để đăng xuất tài khoản. |
| **Xử lý** | * Hủy toàn bộ session của người dùng hiện tại * Đưa trở về chức năng của khách vãng lai. |
| **Xuất** | Hiển thị giao diện như khách vãng lai sau khi đã đăng xuất khỏi hệ thống. |

* **Đặt hàng**
* **Quản Lý Tài Khoản Cá Nhân**
* **Theo Dõi Đơn Hàng và Lịch Sử Mua Sắm**
* **Tính Năng Thanh Toán Linh Hoạt**
* **Quản Lý Thông Tin Ưa Thích**
* **Đánh Giá và Nhận Xét Sản Phẩm**
* **Cập Nhật Thông Báo và Tin Tức**
* **Tùy Chỉnh Thông Tin Cá Nhân**
* **Các Ưu Đãi và Khuyến Mãi Đặc Biệt**
  + 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. |

* **Quản Lý Đơn Hàng**

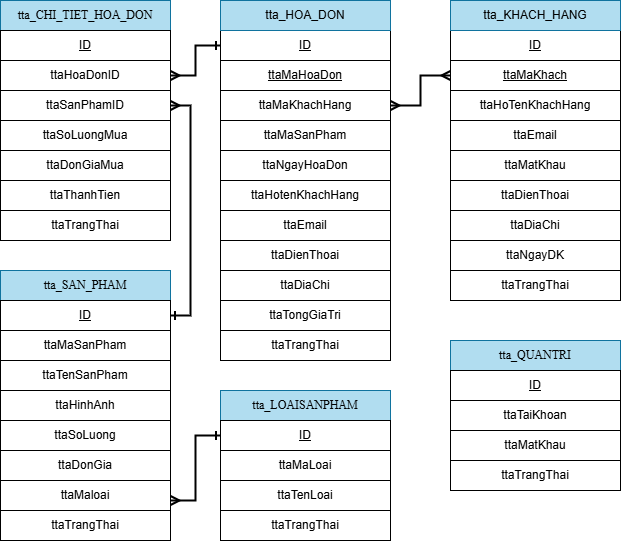
|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc khóa/kích hoạt, xóa các hóa đơn của thành viên. |
| **Nhập** | Chọn lệnh tương ứng của hóa đơn cần thao tác. |
| **Xử lý** | * Với việc xem thông tin hóa đơn: * Hiển thị thông tin của hóa đơn. * Đối với việc xóa hóa đơn: *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

* **Chi tiết hóa đơn khách hàng**

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

## Systeme Designs (Thiết kế hệ thống)

## 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 (**nvk\_quantri**):   
   Lưu trữ thông tin người quản trị viên hệ thống.

**nvk\_quantri**

**Tai\_khoan**

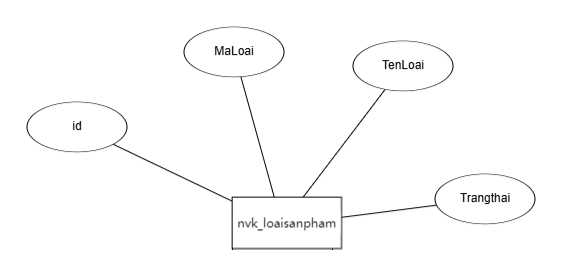
Mat\_khau

Trang\_thai

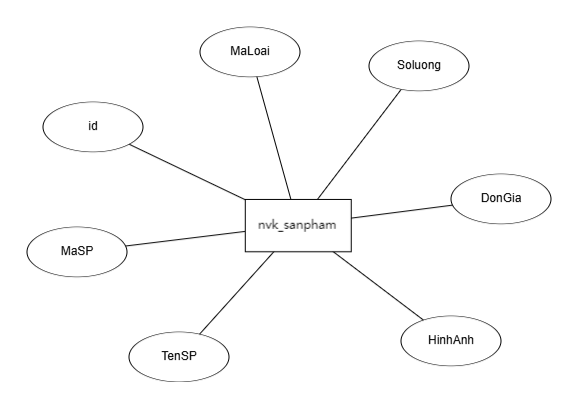
1. Thực thể khách hàng (**nvk\_khachhang**): Lưu trữ thông tin khách hàng khi đăng ký, mua hàng tại hệ thống.



1. Thực thể loại sản phẩm (**nvk\_loaisanpham**) : Lưu trữ thông tin loại sản phẩm khi đăng ký, mua hàng tại hệ thống.



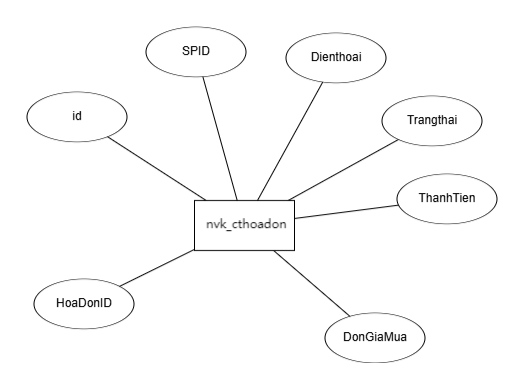
1. Thực thể sản phẩm (**nvk\_sanpham**) : Lưu trữ thông tin sản phẩm khi đăng ký, mua hàng tại hệ thống



1. Thực thể hóa đơn (**nvk\_hoadon**) : Lưu trữ thông tin hóa đơn khi đăng ký, mua hàng tại hệ thống.



1. 6. Thực thể chi tiết hóa đơn (**nvk\_cthoadon**) : Lưu trữ thông tin chi tiết hóa đơn khi đăng ký, mua hàng tại hệ thống.



## 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 **QUAN\_TRI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| nvkTaiKhoan | bigint(20) | No | Primary Key | Tài khoản đăng nhập của quản trị |
| nvkMatKhau | Varchar(255) | No |  | Mật khẩu đăng nhập của quản trị. |
| nvkTrangThai | Tinyint(4) |  |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

1. Bảng **KHACH\_HANG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| **MaKH** | Int | No | Primary Key; Auto\_increment | Mã khách hàng |
| nvkMakhachhang | Varchar(100) | Yes |  | Tên khách hàng |
| nvkHotenkhachhang | Varchar(50) | No | Unique | Tài khoản đăng nhập |
| nvkDienThoai | Varchar(30) | Yes |  | Điện thoại |
| nvkEmail | Varchar(50) | No |  | Hộp thư điện tử |
| Ngay\_sinh | DateTime | Yes |  | Ngày sinh |
| nvkNgayDK | DateTime | Yes | Current | Ngày cập nhật vào hệ thống |
| nvkTrangThai | Tinyint | Yes |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

1. Bảng Mã Sản Phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột (Field) | Kiểu dữ liệu DataType | Null | Mô tả |
| nvkMaloai | varchar(255) | No | Mã loại sản phẩm |
| nvkTenLoai | varchar(255) | No | Tên loại sản phẩm |
| nvkTrangThai | tinyint(1) | No | Trạng thái:  1 - Đang hoạt động 0 - Đang bị khóa |

1. Bảng sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột (Field) | Kiểu dữ liệu DataType | Null | Mô tả q |
| nvkMaSanPham | varchar(255) | No | Mã của sản phẩm |
| nvkTenSanPham | varchar(255) | No | Tên của sản phẩm |
| nvkHinhAnh | varchar(255) | Yes | Ảnh của sản phẩm |
| nvkSoLuong | int(11) | No | Số lượng của sản phẩm |
| nvkDonGia | double(10,2) | No | Đơn giá của sản phẩm |
| nvkMaloai | bigint(20) | No | Mã loại của sản phẩm |
| nvkTrangThai | tinyint(4) | No | Trạng thái:  1 - Đang hoạt động 0 - Đang bị khóa |

1. Bảng Hoá Đơn

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột (Field) | Kiểu dữ liệu DataType | Null | Mô tả |
| nvkMaHoaDon | varchar(255) | No | Mã hoá đơn của khách hàng |
| nvkMakhachhang | bigint(20) | No | Mã của khách hàng |
| nvkNgayHoaDon | date | No | Ngày tạo hoá đơn |
| nvkHotenKhachHang | varchar(255) | No | Tên của khách hàng |
| nvkEmail | varchar(255) | No | Email của khách hàng |
| nvkDienThoai | varchar(255) | No | Điện thoại khách hàng |
| nvkDiaChi | varchar(255) | No | Địa chỉ khách hàng |
| nvkTongGiaTri | double(8,2) | No | Tổng giá trị của khách hàng |
| nvkTrangThai | tinyint(4) | No | Trạng thái:  1 - Đang hoạt động 0 - Đang bị khóa |

1. Bảng CT Hoá Đơn

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột (Field) | Kiểu dữ liệu DataType | Null | Mô tả |
| nvkHoaDonID | int(11) | No | Mã hoá đơn của khách hàng |
| nvkSanPhamID | int(11) | No | Mã sản phẩm của khách hàng |
| nvkSoLuongMua | int(11) | No | Số lượng mua của khách hàng |
| nvkDonGiaMua | double(8,2) | No | Đơn giá mua của khách hàng |
| nvkThanhTien | double(8,2) | No | Tổng tiền của khách hàng |
| nvknvkTrangThai | tinyint(4) | No | Trạng thái:  1 - Đang hoạt động 0 - Đang bị khóa |

## Site map (Cấu trúc trang): Dành cho khách vãng lai

Trang chủ

Sản phẩm

Nhãn hiệu

Tìm kiếm

Giới thiệu

Liên hệ

Hướng dẫn

Đăng nhập

Đăng xuất

Giỏ hàng

## 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

## Sitemap (Cấu trúc trang): Dành cho quản trị viên

Trang chủ Admin

Quản lý Sản phẩm

Quản lý Bình luận

Quản lý Loại

Quản lý Nhãn hiệu

Quản lý Hóa đơn

Quản lý Liên hệ

Quản lý Phản hồi

Quản lý Tin tức

Quản lý Thành viên

Quản lý Quản trị

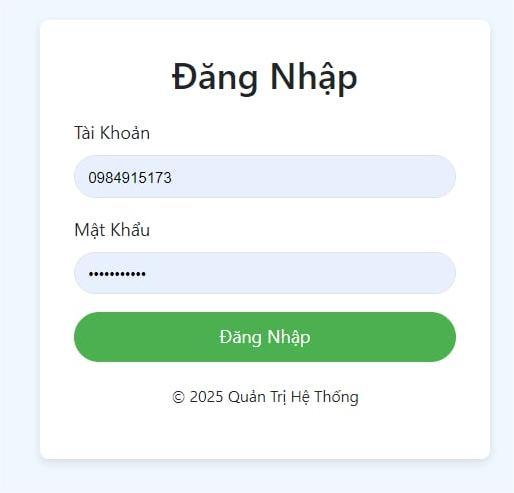
# **CHƯƠNG 4: XÂY DỰNG HỆ THỐNG**

## Một số mã nguồn chương trình

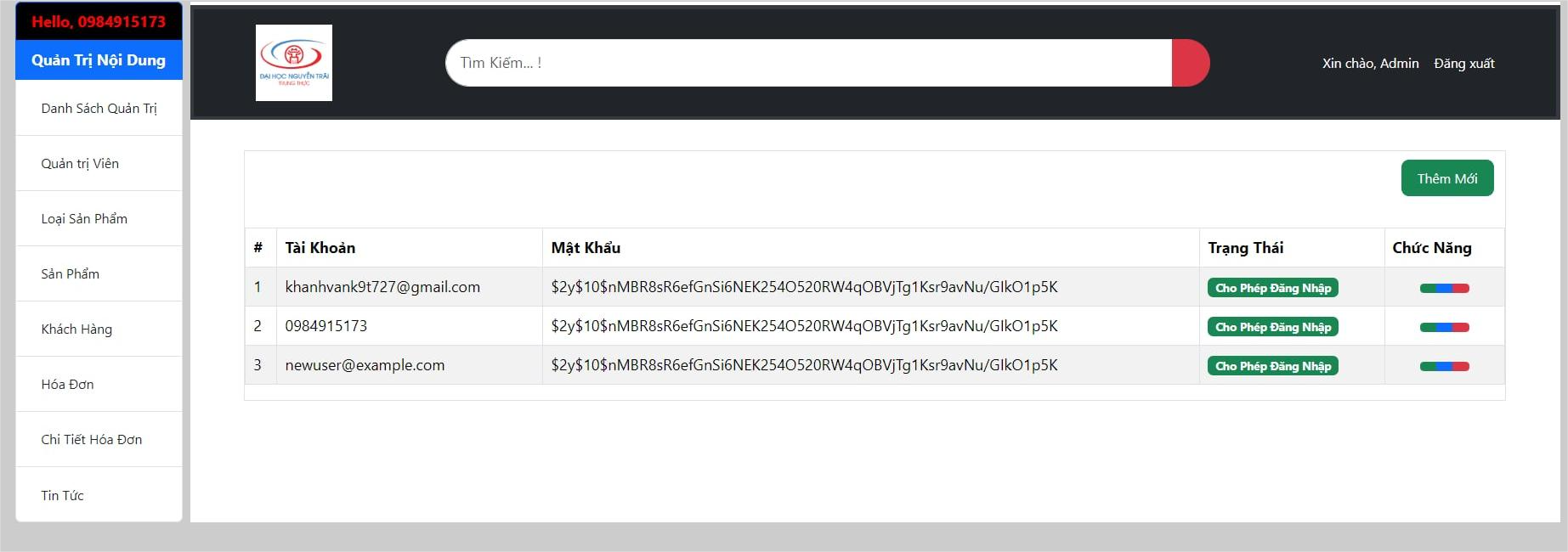
## <https://github.com/KhanChinTuoi/K23CNT3_NguyenVanKhanh_Project01LabGK>

## Một số giao diện người dùng và chức năng cụ thể

## Giao diện trang quản trị

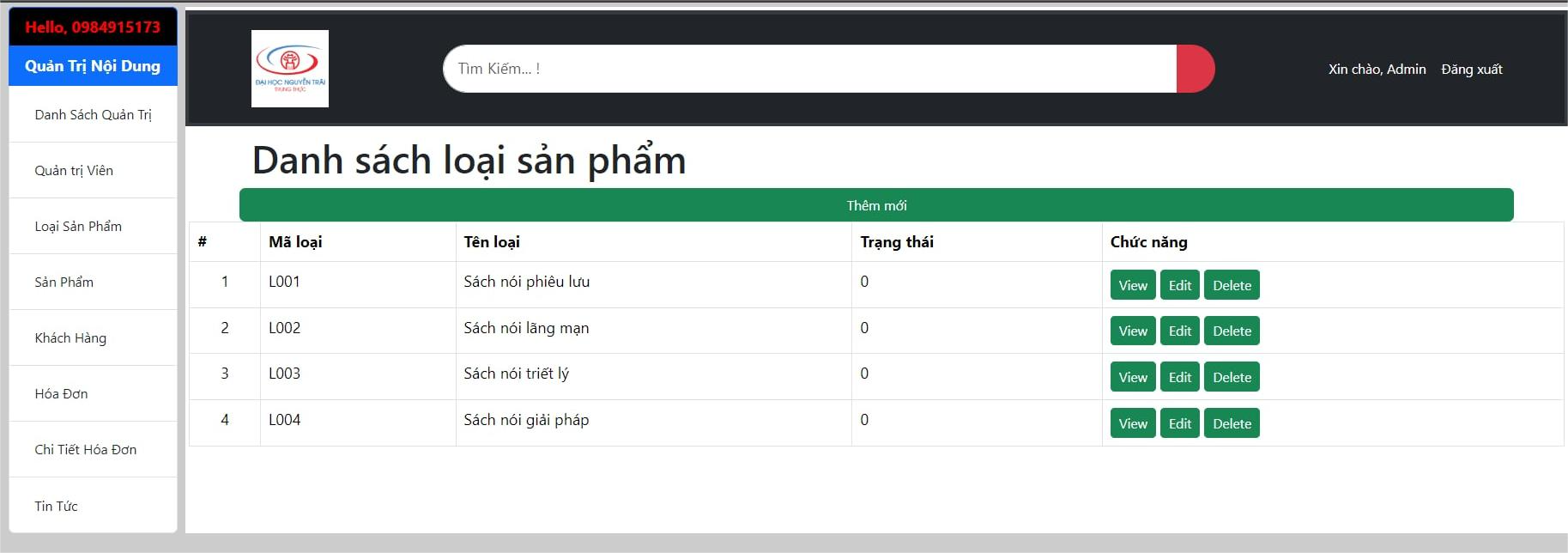


## Giao diện danh sách Admin



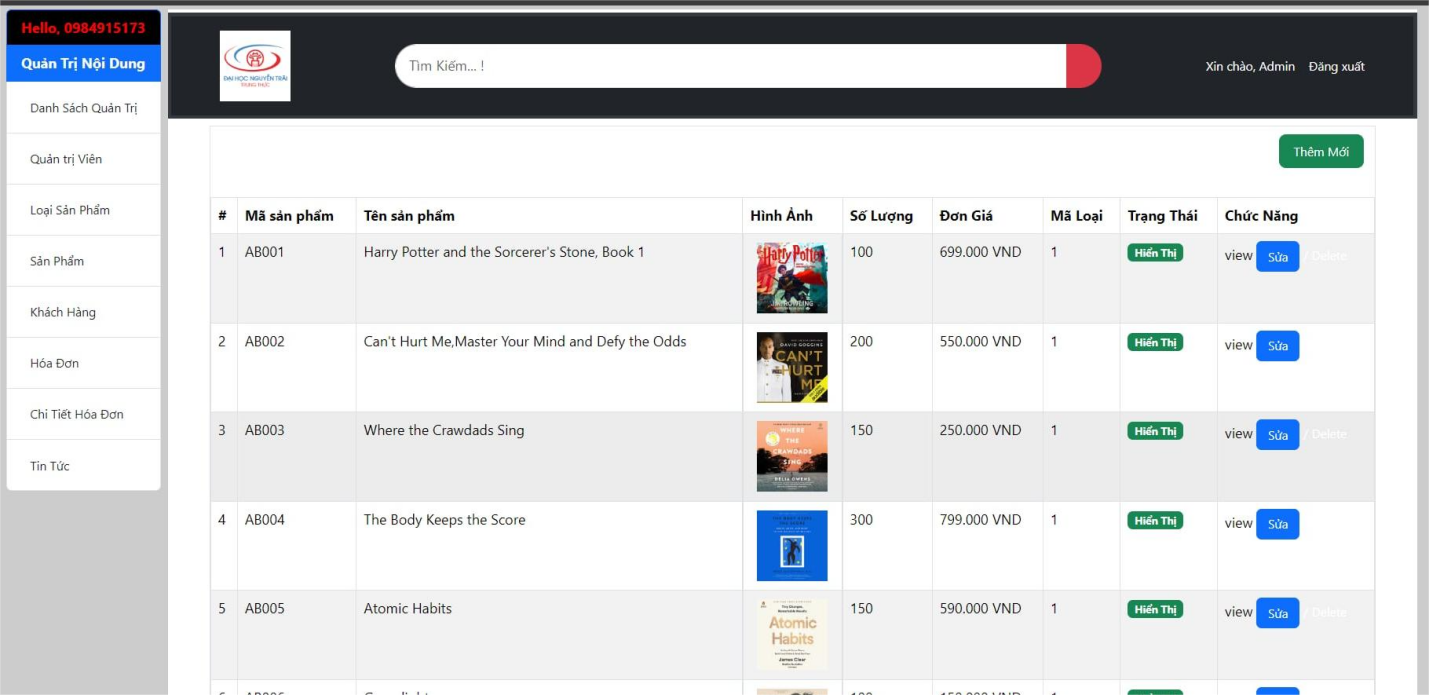
3. Giao diện danh sách loại sản phẩm

Đây là giao diện danh sách loại sản phẩm để xem chi tiết thông tin về loại sản phẩm, quản lý các loại sản phẩm thay đổi tài loại sản phẩm, mã loại, tên loại, trạng thái, hoặc thêm, sửa, xoá theo ý của người dùng lưu vào database để lưu trữ.



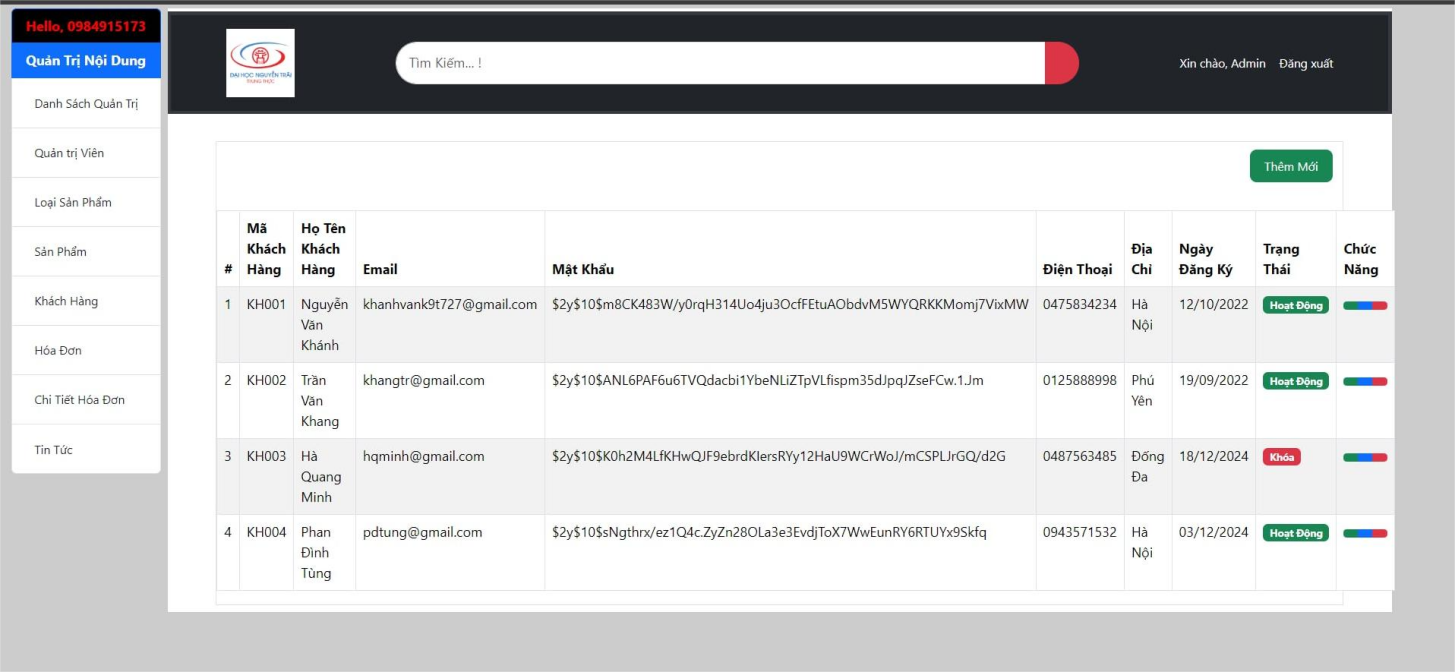
4.Giao diện danh sách sản phẩm

Đây là giao diện danh sách sản phẩm để xem chi tiết thông tin về sản phẩm, quản lý các sản phẩm thay đổi sản phẩm, mã sản phẩm, tên sản phẩm, hình ảnh, số lượng, đơn giá, mã loại, trạng thái, hoặc thêm, sửa, xoá theo ý của người dùng lưu vào database để lưu trữ.



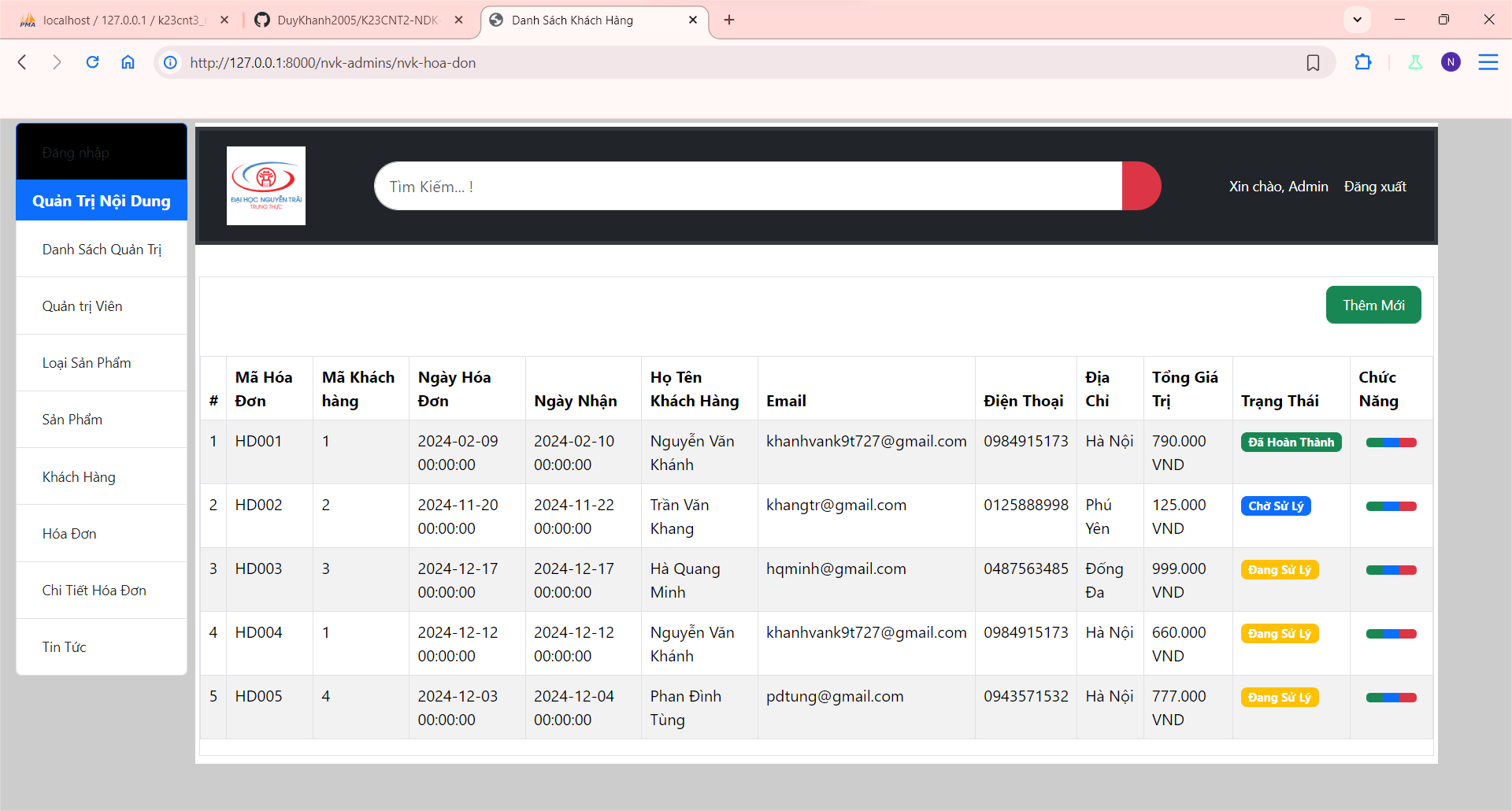
5. Giao diện danh sách khách hàng

Đây là giao diện danh sách khách hàng để xem chi tiết thông tin về khách hàng, quản lý các khách hàng thay đổi mã khách hàng, họ tên khách hàng, email, điện thoại, địa chỉ, ngày đăng kí, trạng thái hoặc thêm, sửa, xoá theo ý của người dùng lưu vào database để lưu trữ.

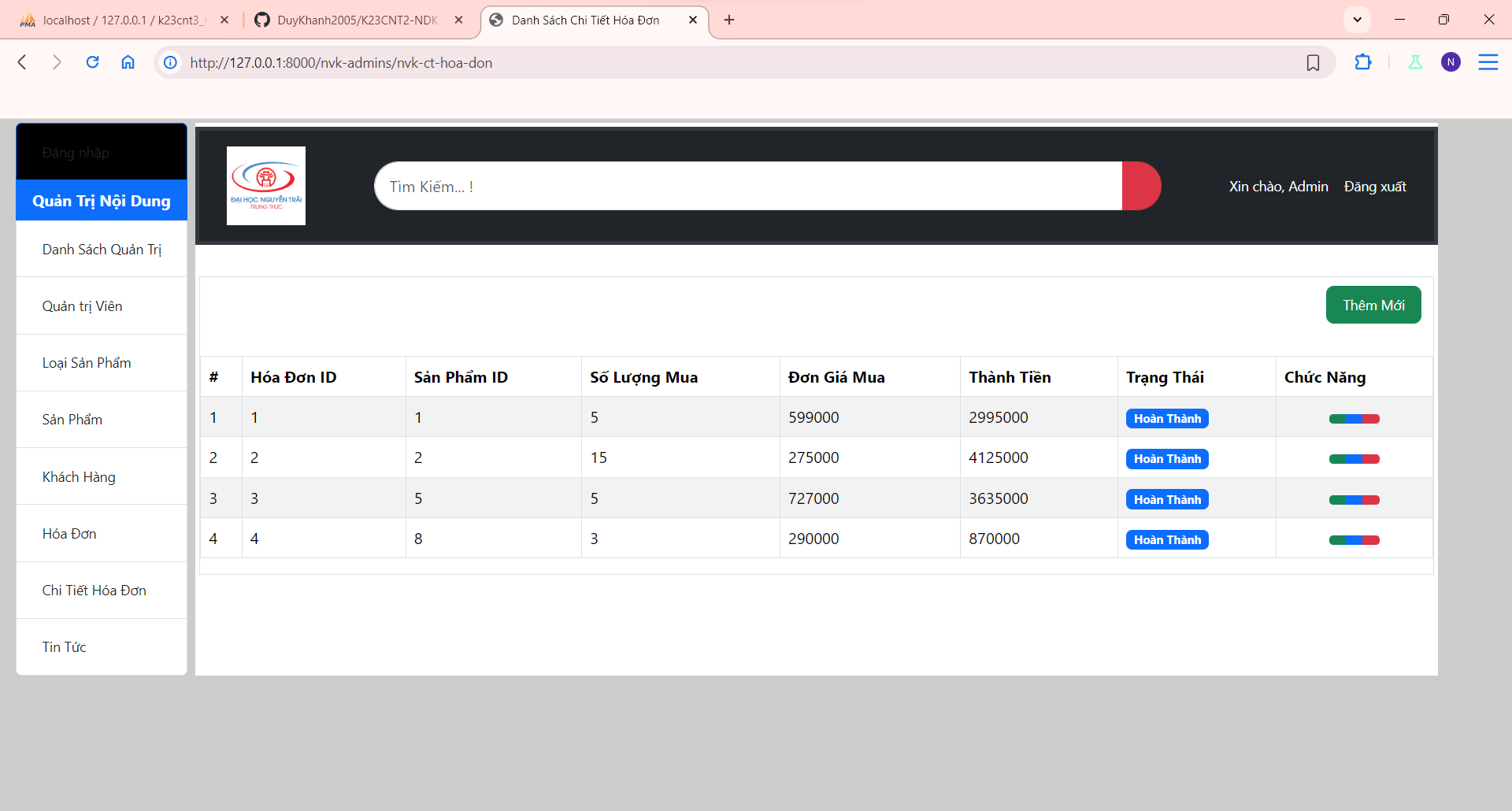


6. Giao diện danh sách hoá đơn

Đây là giao diện danh sách hoá đơn để xem chi tiết thông tin về hoá đơn, quản lý các hoá đơn thay đổi mã hoá đơn, mã khách hàng, ngày hoá đơn, họ tên khách hàng, email, điện thoại, địa chỉ, tổng giá trị, trạng thái hoặc thêm, sửa, xoá theo ý của người dùng lưu vào database để lưu trữ.

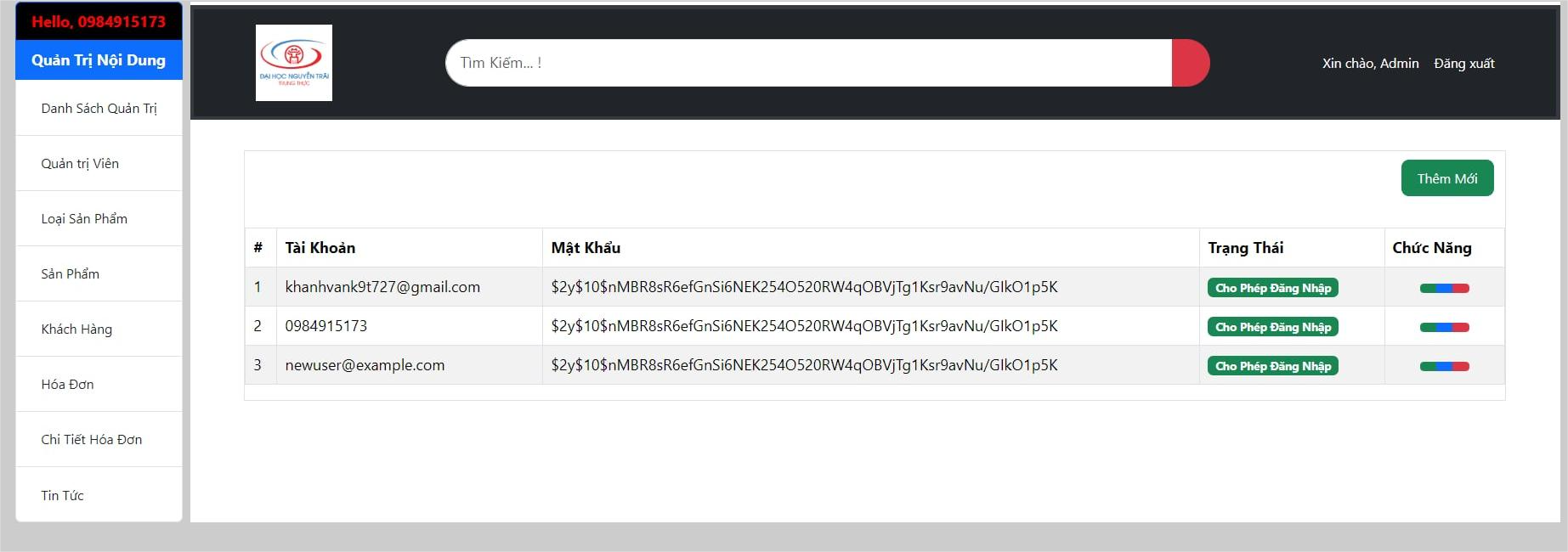


7. Đây là giao diện danh sách chi tiết hoá đơn để xem chi tiết thông tin về chi tiết hoá đơn, quản lý các chi tiết hoá đơn thay đổi Id hoá đơn, Id sản phẩm, số lượng mua, đơn giá mua, thành tiền, trạng thái hoặc thêm, sửa, xoá theo ý của người dùng lưu vào database để lưu trữ.



8.Giao diện đăng xuất

Đây là giao diện đăng xuất sau khi admin nhấn đăng xuất thì admin sẽ trở ra ngoài trang login



5.1. Kết luận

Quá trình làm đồ án chuyên đề với đề tài Fast Food Website 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, bootstrap, Javascript

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:

* + Quản lý người dùng
  + Thêm mới dữ liệu
  + Sửa dữ liệu
  + Xóa dữ liệu
  + Kích hoạt và ngắt kích hoạt dữ liệu

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 của phần mền thân thiện hơn với người dùng và có thêm nhiều tính năng hay hơn cho người dùng.

1. **Task Sheet (bảng kế hoạch công việc)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Giai đoạn** | **Từ ngày … đến….** | **Công việc** | **Tỉ lệ hoàn thành** |
| Chương 1:   (100%) | Từ ngày 23 đến 24 tháng 11-2024 | * Giới thiệu về HTML5 * Giới Thiệu Về CSS3 * Tổng hợp về ngôn ngữ lập trình javascript * Thư viện Jquery * Framework Bootstrap * Tổng quan về hệ quản trị cơ sở dữ liệu MySQL * Giới thiệu về MySQL * Ưu điểm của MySQL * Nhược điểm của MySQL * Tông quát về ngôn ngữ lập trình PHP * Ưu điểm của MySQL * Nhược điểm của MySQL * Framework Laravel | 100% |
| Chương 2:   (100%) | Từ ngày 24 đến 25 tháng 11-2024 | * Đặt vẫn đề * Hệ thống hiện tại * Hệ thống đề xuất * Giới hạn hệ thống * Yêu cầu về phần cứng phần mềm * Yêu cầu tối thiểu * Yêu cầu để nghỉ | 100% |
| Chương 3:   (100%) | Từ ngày 25 đến 26 tháng 11-2024 | * Người dùng của hệ thống * Chức năng của hệ thống * Systeme Designs(Thiết kế hệ thống) * Mô hình thực tế * Thiết kế cơ sở dữ liệu * Đăng Nhập * Đăng xuất | 100% |
| Chương 4:   (100%) | Từ ngày 27 đến 28 tháng 11-2024 | * Xây dựng phần mền * Một số mã nguồng trương trình * Một số giao diện người dùng và chức năng cụ thệ * Giao diện trang chủ * Giao diện trang quản trị * Giao diện danh sách loại sản phẩm * Giao diện danh sách sản phẩm * Giao diện danh sách khách hàng * Giao diên danh sách hóa đơn * Giao diện danh sách chi tiết hóa đơn | 100% |

* 1. **Checklist (bảng kiểm tra chức năng)**
     1. **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 |

* + 1. **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:**

**Website:**

1. <https://www.w3schools.com/>
2. <https://support.microsoft.com/vi-vn/topic/31846479-c656-f2a4-bc24-c9803a97e62c>
3. <https://topdev.vn/blog/sql-server-la-gi/>