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

***ĐỀ TÀI***

**XÂY DỰNG WEB BÁN NƯỚC HOA**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:** | **Phạm Thị Thu Huyền** |
| **Mã sinh viên:** | **2210900030** |
| **Lớp:** | **K22CNT4** |
| **Khóa:** | **K22** |

**@FITNTU-PJ** <<tháng>>/<<Năm>>

**MỤC LỤC**

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

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

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

[**CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT** 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** 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:** 15](#_heading=h.upglbi)

[**PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG** 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** 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** 30](#_heading=h.j8sehv)

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

[5.2.](#_heading=h.1idq7dh) TaskSheet (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** 34](#_heading=h.3gnlt4p)

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

# **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 HTML

**HTML** (HyperText Markup Language) là ngôn ngữ đánh dấu được sử dụng để tạo các trang web và ứng dụng web. Đây là ngôn ngữ cơ bản của web, giúp cấu trúc nội dung trang web thông qua các thẻ (tags). Các thẻ HTML được sử dụng để định nghĩa các phần tử như tiêu đề, đoạn văn, liên kết, hình ảnh, bảng biểu, biểu mẫu và nhiều thành phần khác trên trang web.

**Các đặc điểm chính của HTML:**

**Ngôn ngữ đánh dấu**: HTML sử dụng các thẻ để đánh dấu và phân loại nội dung. Mỗi thẻ bao gồm một tên thẻ và có thể có các thuộc tính kèm theo.

Ví dụ: <p> để tạo một đoạn văn bản, <a> để tạo một liên kết.

**Cấu trúc của tài liệu HTML**:

* + **Doctype**: Mở đầu tài liệu HTML thường là một khai báo Doctype để chỉ định phiên bản HTML đang được sử dụng, như <!DOCTYPE html>.
  + **Thẻ <html>**: Bao bọc toàn bộ nội dung của trang.
  + **Thẻ <head>**: Chứa thông tin về tài liệu như tiêu đề trang, liên kết tới tập tin CSS, meta tags (thông tin về mã hóa ký tự, từ khóa SEO, v.v.).
  + **Thẻ <body>**: Chứa nội dung hiển thị trên trình duyệt như văn bản, hình ảnh, video, liên kết, biểu mẫu...

**Đa nền tảng**: HTML là ngôn ngữ độc lập nền tảng, có thể chạy trên mọi trình duyệt và hệ điều hành.

* + **Liên kết đa phương tiện**: HTML hỗ trợ nhúng các yếu tố đa phương tiện như âm thanh, video, hình ảnh, cũng như tích hợp với các ngôn ngữ khác như CSS (Cascading Style Sheets) để định kiểu hoặc JavaScript để thêm tương tác.

HTML là ngôn ngữ cơ bản nhất mà mọi nhà phát triển web cần nắm vững để tạo nên các trang web tương tác và thân thiện với người dùng.

## Giới thiệu về CSS3

**CSS3** (Cascading Style Sheets level 3) là phiên bản mới nhất của CSS, một ngôn ngữ được sử dụng để mô tả cách hiển thị các thành phần trên trang web được tạo bằng HTML. Nếu HTML cung cấp cấu trúc cho trang web, thì CSS là ngôn ngữ tạo nên giao diện và định dạng, giúp làm cho trang web trở nên thẩm mỹ và dễ nhìn hơn.

**Các đặc điểm chính của CSS3:**

**Phân tầng và thừa kế**:

* + CSS có khả năng phân tầng (cascading), cho phép các quy tắc phong cách được kế thừa hoặc ghi đè. Nếu có nhiều quy tắc áp dụng cho một phần tử, CSS sẽ xác định thứ tự ưu tiên dựa trên tính đặc hiệu (specificity) và thứ tự của các quy tắc.

**Module hóa**:

* + **CSS3** được chia thành nhiều module (mô-đun) độc lập, mỗi mô-đun tập trung vào một khía cạnh cụ thể của trang web. Các module phổ biến gồm:
    - **Selectors** (bộ chọn)
    - **Box Model** (mô hình hộp)
    - **Backgrounds & Borders** (nền và viền)
    - **Text Effects** (hiệu ứng văn bản)
    - **2D/3D Transformations** (chuyển đổi 2D/3D)
    - **Animations** (hoạt hình)

**Các tính năng mới trong CSS3**:

* + **Gradients (màu chuyển tiếp)**: Cho phép tạo hiệu ứng chuyển tiếp màu sắc (linear hoặc radial).
  + **Border-radius**: Tạo viền bo tròn cho các phần tử mà không cần phải dùng hình ảnh.
  + **Shadow (bóng đổ)**: Thêm bóng đổ cho các phần tử và văn bản (box-shadow và text-shadow).
  + **Transitions (chuyển tiếp)**: Tạo hiệu ứng mượt mà khi thay đổi trạng thái của một phần tử.
  + **Animations (hoạt hình)**: Cho phép tạo các hiệu ứng hoạt hình phức tạp mà không cần sử dụng JavaScript.
  + **Media Queries (truy vấn phương tiện)**: Cho phép thay đổi phong cách trang web dựa trên kích thước màn hình, giúp xây dựng các trang web đáp ứng (responsive design).

**Responsive Design**:

* + CSS3 hỗ trợ **Media Queries**, cho phép thiết kế các trang web có thể tự điều chỉnh kích thước và giao diện để phù hợp với mọi loại thiết bị như điện thoại, máy tính bảng, hoặc màn hình lớn.
  + Ví dụ: Thiết lập kích thước phần tử theo phần trăm thay vì cố định để trang web linh hoạt hơn khi hiển thị trên nhiều kích thước màn hình.

**Lợi ích của CSS3:**

* + **Tiết kiệm thời gian**: Giúp tách biệt phần giao diện với nội dung HTML, dễ quản lý và tái sử dụng.
  + **Tăng tính tương tác**: Các hiệu ứng và hoạt hình của CSS3 giúp tạo trải nghiệm người dùng thú vị hơn.
  + **Thiết kế phản hồi**: CSS3 cho phép tạo ra các trang web phản hồi linh hoạt, thích ứng với các kích thước màn hình khác nhau.

## Tổng quan về ngôn ngữ lập trình Javascript

**JavaScript** là một ngôn ngữ lập trình kịch bản (scripting language) phổ biến, được sử dụng chủ yếu để phát triển các ứng dụng web và tạo ra các tương tác động trên các trang web. Ban đầu, JavaScript được phát triển để chạy trên trình duyệt web của người dùng (client-side), nhưng ngày nay nó cũng có thể được chạy trên máy chủ (server-side) nhờ vào môi trường như **Node.js**.

**Đặc điểm chính của JavaScript:**

* **Ngôn ngữ động và hướng đối tượng**: JavaScript hỗ trợ lập trình hướng đối tượng thông qua cơ chế **prototypes** thay vì các class truyền thống như trong Java hoặc C++.
* **Đa nền tảng**: JavaScript có thể chạy trên hầu hết các trình duyệt hiện đại như Chrome, Firefox, Safari, Internet Explorer, và các hệ điều hành như Windows, macOS, Linux.
* **Chạy trực tiếp trên trình duyệt**: Khi người dùng truy cập trang web, mã JavaScript được tải xuống và thực thi trực tiếp trên trình duyệt mà không cần bất kỳ plugin nào.
* **Hỗ trợ sự kiện (Event-driven)**: JavaScript cung cấp các cơ chế để xử lý sự kiện (event), như click chuột, gõ phím, hay tải trang, giúp tạo ra các tương tác động và linh hoạt.

**Tính năng nổi bật của JavaScript:**

* **Tương tác động trên trang web**: JavaScript cho phép thực hiện các thao tác thay đổi nội dung của trang web mà không cần tải lại toàn bộ trang, như thay đổi nội dung HTML, hiển thị hoặc ẩn các phần tử, hay tương tác với người dùng qua các hộp thoại (popup).
* **DOM Manipulation (Thao tác DOM)**: JavaScript cho phép thao tác với **Document Object Model (DOM)** của trang web, giúp thay đổi cấu trúc của trang web khi trang đã được tải. DOM là đại diện cây cấu trúc của trang web, và JavaScript có thể thêm, xóa, hoặc thay đổi các phần tử trong cây DOM.
* **AJAX (Asynchronous JavaScript and XML)**: JavaScript có thể gửi yêu cầu tới máy chủ và nhận dữ liệu mà không cần tải lại toàn bộ trang. AJAX là công nghệ giúp cải thiện trải nghiệm người dùng bằng cách cho phép tải nội dung trang một cách linh hoạt mà không làm gián đoạn hoạt động của trang web.
* **Xử lý bất đồng bộ (Asynchronous)**: JavaScript hỗ trợ lập trình bất đồng bộ thông qua **callbacks**, **Promises**, và **async/await**, giúp quản lý các tác vụ như gọi API hoặc xử lý dữ liệu không đồng bộ.

**Sử dụng JavaScript ở phía client (Client-side JavaScript):**

* **Tạo tương tác động**: JavaScript giúp trang web trở nên linh hoạt và tương tác với người dùng hơn, như hiển thị hộp thoại cảnh báo, xác minh thông tin người dùng nhập vào biểu mẫu (form validation), hoặc thay đổi nội dung trang dựa trên hành vi của người dùng.
* **Xử lý sự kiện**: JavaScript có thể lắng nghe và xử lý các sự kiện do người dùng thực hiện như di chuột, nhấn phím, nhấn nút, hoặc cuộn trang.
* **Thao tác trên DOM**: JavaScript có thể truy cập và chỉnh sửa các phần tử HTML của trang web, từ đó thay đổi nội dung hiển thị, định dạng hoặc phong cách của các phần tử.

**Sử dụng JavaScript ở phía máy chủ (Server-side JavaScript):**

* **Node.js** là nền tảng giúp JavaScript có thể chạy ở phía máy chủ. Với Node.js, JavaScript có thể thực hiện các tác vụ như đọc/ghi tệp, tương tác với cơ sở dữ liệu, và xử lý các yêu cầu HTTP từ trình duyệt.
* **Lợi ích**: Khi sử dụng JavaScript ở cả client-side và server-side, các nhà phát triển có thể dùng chung một ngôn ngữ lập trình cho toàn bộ ứng dụng, giúp đơn giản hóa quá trình phát triển và bảo trì ứng dụng.

**Ưu điểm của JavaScript:**

* **Dễ học và dễ sử dụng**: JavaScript có cú pháp tương đối đơn giản và dễ tiếp cận, đặc biệt là với những người mới bắt đầu học lập trình web.
* **Tương tác thời gian thực**: JavaScript có thể thực hiện các tác vụ mà không cần tải lại trang, giúp cải thiện trải nghiệm người dùng và tạo ra các ứng dụng web tương tác như các trò chuyện trực tuyến (chat), công cụ chỉnh sửa tài liệu trực tiếp...
* **Khả năng mở rộng**: JavaScript có thể được mở rộng thông qua các thư viện và framework như **jQuery**, **React**, **Angular**, và **Vue.js**, giúp giải quyết các vấn đề phức tạp hơn trong phát triển giao diện người dùng.
* **Hỗ trợ cộng đồng lớn**: JavaScript là ngôn ngữ lập trình phổ biến nhất hiện nay, với hàng triệu nhà phát triển, tài liệu phong phú và rất nhiều công cụ hỗ trợ.

**Nhược điểm của JavaScript:**

* **Hiệu suất**: JavaScript chạy trên máy của người dùng nên có thể làm chậm hiệu suất nếu mã không được tối ưu hóa hoặc nếu người dùng sử dụng thiết bị có cấu hình yếu.
* **Bảo mật**: Vì JavaScript chạy ở phía client, mã nguồn có thể dễ dàng bị xem hoặc chỉnh sửa bởi người dùng, dẫn đến các vấn đề về bảo mật. Các cuộc tấn công như **Cross-site Scripting (XSS)** có thể lợi dụng JavaScript để chèn mã độc vào trang web.
* **Khác biệt giữa các trình duyệt**: Mặc dù đã có sự chuẩn hóa, vẫn có những khác biệt nhỏ giữa các trình duyệt trong cách xử lý JavaScript, dẫn đến việc cần kiểm tra và tối ưu hóa mã cho nhiều nền tảng.

**JavaScript frameworks và libraries:**

* **jQuery**: Thư viện JavaScript đơn giản hóa việc thao tác DOM, xử lý sự kiện và tương tác AJAX.
* **React**: Thư viện mạnh mẽ phát triển bởi Facebook để xây dựng giao diện người dùng động, tập trung vào việc xây dựng các thành phần UI có thể tái sử dụng.
* **Angular**: Một framework phát triển bởi Google, sử dụng để xây dựng các ứng dụng web phức tạp theo mô hình MVC (Model-View-Controller).
* **Vue.js**: Một framework nhẹ và linh hoạt giúp xây dựng giao diện người dùng tương tác.

## Framework Bootstrap

**Bootstrap** là một framework mã nguồn mở, được phát triển bởi Twitter, giúp xây dựng giao diện web nhanh chóng và dễ dàng. Bootstrap cung cấp sẵn các thành phần giao diện người dùng như **bố cục lưới**, **nút**, **biểu mẫu**, **thanh điều hướng**, và nhiều thành phần khác. Với các tính năng như tính tương thích trình duyệt cao và thiết kế **responsive** (phản hồi linh hoạt), Bootstrap giúp nhà phát triển tạo ra các trang web đẹp và dễ sử dụng mà không cần phải viết quá nhiều mã CSS và JavaScript từ đầu.

**Đặc điểm chính của Bootstrap:**

* **Thiết kế responsive**: Bootstrap sử dụng hệ thống lưới (grid system) giúp trang web có thể tự động điều chỉnh bố cục và kích thước dựa trên kích thước màn hình của thiết bị, từ điện thoại di động, máy tính bảng đến máy tính để bàn.
* **Tính nhất quán**: Bootstrap cung cấp các thành phần giao diện nhất quán về phong cách, giúp tạo ra các trang web có giao diện đồng nhất. Điều này giúp giảm thiểu sự khác biệt về giao diện giữa các trình duyệt khác nhau.
* **Thư viện phong phú các thành phần giao diện**: Bootstrap cung cấp sẵn nhiều thành phần như các nút, biểu mẫu, thanh điều hướng, bảng, thẻ, mô-đun (modal), thông báo, thanh tiến trình (progress bar), v.v. giúp giảm thiểu công việc thiết kế giao diện từ đầu.
* **Tùy biến dễ dàng**: Người dùng có thể tùy chỉnh giao diện Bootstrap bằng cách ghi đè các lớp CSS mặc định hoặc chỉnh sửa tập tin SASS/LESS để phù hợp với thiết kế và nhu cầu của mình.

**Các tính năng nổi bật của Bootstrap:**

* **Hệ thống lưới (Grid system)**: Bootstrap cung cấp hệ thống lưới 12 cột giúp dễ dàng xây dựng các bố cục phức tạp. Hệ thống lưới này tự động điều chỉnh bố cục theo kích thước màn hình của thiết bị.
  + **Ví dụ**: Bootstrap sử dụng các lớp như .col-md-6, .col-lg-4 để chia các phần tử thành các cột tương ứng trên màn hình có kích thước trung bình (md) hoặc lớn (lg).
* **Các lớp tiện ích (Utility classes)**: Bootstrap có sẵn các lớp tiện ích để định dạng nhanh các thành phần, như lớp cho việc căn chỉnh văn bản, đệm (padding), lề (margin), ẩn hoặc hiển thị phần tử dựa trên kích thước màn hình.
  + Ví dụ: Lớp text-center giúp căn giữa văn bản, hoặc lớp d-none giúp ẩn phần tử trên trang.
* **Thành phần JavaScript tích hợp sẵn**: Bootstrap đi kèm với nhiều thành phần JavaScript được xây dựng sẵn như **mô-đun** (modal), **carousel** (trình chiếu hình ảnh), **dropdown** (thả xuống), và nhiều tính năng tương tác khác, giúp tạo ra các trang web động mà không cần viết nhiều mã JavaScript thủ công.

### ****Cách sử dụng Bootstrap****:

* Để sử dụng Bootstrap, có thể chèn trực tiếp các tệp CSS và JS từ **CDN** (Content Delivery Network) hoặc tải xuống các tệp nguồn và nhúng vào dự án của mình.

**Hệ thống lưới (Grid System) trong Bootstrap:**

* Bootstrap sử dụng hệ thống lưới **12 cột**, nơi mỗi phần tử có thể chiếm một hoặc nhiều cột tùy thuộc vào kích thước màn hình. Hệ thống lưới này giúp bố trí các phần tử dễ dàng và có khả năng tự điều chỉnh theo kích thước thiết bị.

**Các kích thước lưới theo màn hình**:

* + **.col-**: Sử dụng cho tất cả các thiết bị.
  + **.col-sm-**: Dành cho thiết bị có kích thước nhỏ (≥ 576px).
  + **.col-md-**: Dành cho thiết bị có kích thước trung bình (≥ 768px).
  + **.col-lg-**: Dành cho thiết bị có kích thước lớn (≥ 992px).
  + **.col-xl-**: Dành cho thiết bị có kích thước rất lớn (≥ 1200px).

**Ưu điểm của Bootstrap:**

* **Nhanh chóng và dễ sử dụng**: Bootstrap giúp tiết kiệm thời gian phát triển vì nó cung cấp sẵn nhiều thành phần giao diện và phong cách CSS phổ biến.
* **Thiết kế responsive**: Dễ dàng tạo các trang web tương thích với mọi thiết bị nhờ hệ thống lưới linh hoạt.
* **Cộng đồng lớn**: Bootstrap có một cộng đồng nhà phát triển mạnh mẽ và tài liệu phong phú, giúp dễ dàng tìm thấy hỗ trợ hoặc tài liệu khi gặp vấn đề.

**Nhược điểm của Bootstrap:**

* **Thiếu tính độc đáo**: Nếu không tùy chỉnh, các trang web sử dụng Bootstrap có thể trông khá giống nhau vì chúng sử dụng các thành phần giao diện mặc định.
* **Kích thước lớn**: Nếu không tùy chỉnh, Bootstrap có thể bao gồm nhiều tính năng không cần thiết, dẫn đến tải trang chậm.
* **Quá nhiều lớp CSS**: Đôi khi việc sử dụng nhiều lớp CSS từ Bootstrap có thể làm mã HTML trở nên khó đọc và phức tạp.

### ****Kết luận****:

Bootstrap là một công cụ mạnh mẽ và tiện dụng cho việc phát triển giao diện web hiện đại. Nó giúp tiết kiệm thời gian, dễ sử dụng và đặc biệt hữu ích cho những ai mới bắt đầu lập trình web. Tuy nhiên, để tạo ra các trang web độc đáo và tối ưu, người phát triển cần biết cách tùy chỉnh Bootstrap một cách hợp lý.

## Tổng quan về hệ quản trị cơ sở đữ liệu SQL Server

## Giới thiệu về SQL Server

**SQL Server** là một hệ quản trị cơ sở dữ liệu quan hệ (**RDBMS - Relational Database Management System**) do Microsoft phát triển. SQL Server được sử dụng rộng rãi trong các tổ chức để quản lý và xử lý dữ liệu. Nó cho phép lưu trữ, truy xuất, và quản lý lượng lớn thông tin theo cách có tổ chức. SQL Server sử dụng **SQL (Structured Query Language)** làm ngôn ngữ chính để thực hiện các thao tác truy vấn và quản lý dữ liệu.

**Các đặc điểm chính của SQL Server:**

* **Quản lý dữ liệu quan hệ**: SQL Server lưu trữ dữ liệu dưới dạng các bảng (tables), trong đó các bảng có thể liên kết với nhau thông qua các khóa (keys) và ràng buộc (constraints).
* **Khả năng mở rộng**: SQL Server có thể mở rộng từ những hệ thống nhỏ đến những ứng dụng doanh nghiệp lớn với khối lượng dữ liệu khổng lồ.
* **Bảo mật**: SQL Server hỗ trợ nhiều tính năng bảo mật như quản lý người dùng và quyền truy cập, mã hóa dữ liệu, và kiểm soát đăng nhập.
* **Hỗ trợ đa nền tảng**: Các phiên bản mới nhất của SQL Server hỗ trợ chạy trên cả Windows và Linux, cung cấp tính linh hoạt trong triển khai.
* **Khả năng khôi phục và dự phòng dữ liệu**: SQL Server có các tính năng như sao lưu dữ liệu tự động, phục hồi khi hệ thống bị lỗi, và sao lưu tại thời điểm (point-in-time recovery), giúp đảm bảo tính toàn vẹn và an toàn của dữ liệu.

**Các thành phần quan trọng của SQL Server:**

* **Database Engine**: Là thành phần cốt lõi chịu trách nhiệm xử lý, lưu trữ và quản lý dữ liệu. Nó xử lý các truy vấn, thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên dữ liệu, và quản lý các giao dịch (transactions).
* **SQL Server Agent**: Dùng để lên lịch và quản lý các tác vụ tự động (automated tasks) như sao lưu dữ liệu, bảo trì cơ sở dữ liệu, hoặc chạy các công việc theo lịch trình.
* **SQL Server Management Studio (SSMS)**: Là công cụ giao diện đồ họa, giúp quản trị viên cơ sở dữ liệu (DBA) có thể dễ dàng quản lý cơ sở dữ liệu, viết truy vấn SQL, và thực hiện các tác vụ liên quan.
* **SQL Profiler**: Là công cụ giúp ghi lại các sự kiện SQL, giúp theo dõi và tối ưu hóa hiệu suất của cơ sở dữ liệu.
* **Integration Services (SSIS)**: Dùng để tích hợp dữ liệu, di chuyển dữ liệu từ các nguồn khác nhau vào SQL Server.
* **Analysis Services (SSAS)**: Dùng cho phân tích dữ liệu và xây dựng các ứng dụng phân tích nâng cao.
* **Reporting Services (SSRS)**: Dùng để tạo và quản lý các báo cáo, cung cấp cho người dùng khả năng tạo báo cáo từ dữ liệu trong SQL Server.

**Các phiên bản của SQL Server:**

SQL Server có nhiều phiên bản khác nhau, mỗi phiên bản hướng tới các nhu cầu sử dụng và quy mô khác nhau:

* **SQL Server Express**: Phiên bản miễn phí, với các tính năng cơ bản, thường được sử dụng cho các ứng dụng nhỏ hoặc mục đích học tập.
* **SQL Server Standard**: Phiên bản trung bình, cung cấp đầy đủ các tính năng cơ bản và một số tính năng nâng cao, phù hợp với doanh nghiệp vừa và nhỏ.
* **SQL Server Enterprise**: Phiên bản cao cấp với tất cả các tính năng, phù hợp với các doanh nghiệp lớn yêu cầu khả năng mở rộng và tính sẵn sàng cao.
* **SQL Server Web Edition**: Phiên bản tối ưu hóa cho các ứng dụng web.
* **SQL Server Developer**: Phiên bản cung cấp tất cả các tính năng của Enterprise, nhưng chỉ được sử dụng cho mục đích phát triển và thử nghiệm, không được sử dụng trong môi trường sản xuất.

**Các tính năng chính của SQL Server:**

* **Transaction Management (Quản lý giao dịch)**: SQL Server hỗ trợ **ACID transactions**, đảm bảo tính nhất quán, an toàn, và tính toàn vẹn của dữ liệu trong suốt quá trình xử lý.
* **Concurrency Control (Kiểm soát đồng thời)**: SQL Server sử dụng cơ chế khóa và phiên bản để quản lý các giao dịch đồng thời, ngăn ngừa xung đột dữ liệu.
* **Full-Text Search**: Hỗ trợ tìm kiếm toàn văn trên các cột văn bản, giúp thực hiện các truy vấn tìm kiếm phức tạp một cách hiệu quả.
* **Tính năng mã hóa**: SQL Server cung cấp nhiều cơ chế mã hóa để bảo vệ dữ liệu, như mã hóa cột, mã hóa tệp tin dữ liệu, mã hóa giao tiếp giữa máy chủ và ứng dụng.
* **Tính năng Always On Availability Groups**: Cung cấp khả năng sao lưu và khôi phục nhanh chóng trong trường hợp hệ thống gặp sự cố, giúp đảm bảo tính sẵn sàng cao của ứng dụng.

## Ưu điểm của SQL Server

**Ưu điểm của SQL Server:**

* **Tính dễ sử dụng**: SQL Server có giao diện quản trị đồ họa mạnh mẽ, dễ sử dụng, phù hợp với cả người dùng mới và các chuyên gia.
* **Tích hợp tốt với hệ sinh thái Microsoft**: SQL Server tích hợp dễ dàng với các sản phẩm khác của Microsoft như **Windows Server**, **Microsoft Azure**, **Power BI**, và **Excel**.
* **Khả năng bảo mật cao**: SQL Server cung cấp nhiều tính năng bảo mật nâng cao, giúp bảo vệ dữ liệu khỏi các cuộc tấn công hoặc truy cập trái phép.
* **Tính năng dự phòng và khôi phục mạnh mẽ**: SQL Server có khả năng sao lưu, khôi phục và tái thiết lập hệ thống trong trường hợp xảy ra sự cố.

## Nhược điểm của SQL Server

**Nhược điểm của SQL Server:**

* **Chi phí cao**: Phiên bản SQL Server Enterprise có chi phí cao, đặc biệt đối với các doanh nghiệp nhỏ hoặc cá nhân sử dụng.
* **Yêu cầu phần cứng**: SQL Server yêu cầu cấu hình phần cứng mạnh mẽ để chạy hiệu quả, đặc biệt khi xử lý dữ liệu lớn và các hệ thống phức tạp.
* **Tích hợp khó với các hệ thống không phải của Microsoft**: Mặc dù SQL Server hỗ trợ Linux, nhưng việc tích hợp tốt nhất vẫn với các hệ thống và phần mềm thuộc hệ sinh thái của Microsoft.

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

## Giới thiệu về nền tảng công nghệ .NET

**.NET** là một nền tảng phát triển phần mềm (software development framework) được Microsoft phát triển nhằm hỗ trợ xây dựng, triển khai và chạy các ứng dụng trên nhiều loại thiết bị và hệ điều hành khác nhau. Với khả năng hỗ trợ đa ngôn ngữ, đa nền tảng và đa thiết bị, .NET là một trong những nền tảng phổ biến được sử dụng để phát triển các ứng dụng doanh nghiệp, ứng dụng web, ứng dụng di động, và các dịch vụ backend.

**Lịch sử và phiên bản**

Nền tảng .NET ban đầu được ra mắt vào năm 2002 với tên gọi **.NET Framework**. Ban đầu, nó chỉ hỗ trợ phát triển ứng dụng trên hệ điều hành Windows. Tuy nhiên, để đáp ứng nhu cầu phát triển đa nền tảng, Microsoft đã phát triển và ra mắt **.NET Core** vào năm 2016. Đây là một phiên bản mã nguồn mở của .NET với khả năng chạy trên các hệ điều hành như Windows, Linux, và macOS.

Vào năm 2020, Microsoft hợp nhất **.NET Framework** và **.NET Core** thành một nền tảng duy nhất gọi là **.NET 5**, và từ đó các phiên bản tiếp theo như **.NET 6**, **.NET 7** được phát triển với các cải tiến về hiệu suất, tính năng và hỗ trợ đa nền tảng.

**Các thành phần chính của .NET**

Nền tảng .NET bao gồm nhiều thành phần và thư viện khác nhau để hỗ trợ các loại ứng dụng khác nhau. Các thành phần chính bao gồm:

**a. Common Language Runtime (CLR)**

CLR là môi trường chạy cho các ứng dụng .NET, chịu trách nhiệm quản lý việc thực thi mã, quản lý bộ nhớ, bảo mật, xử lý ngoại lệ và quản lý luồng. Nó cung cấp cơ chế thực thi chương trình một cách an toàn và tối ưu.

**b. Base Class Library (BCL)**

BCL là tập hợp các thư viện tiêu chuẩn mà các ứng dụng .NET có thể sử dụng để thực hiện các chức năng cơ bản như xử lý tệp, thao tác dữ liệu, quản lý giao tiếp mạng, và nhiều chức năng khác.

**c. .NET Languages**

.NET hỗ trợ nhiều ngôn ngữ lập trình, phổ biến nhất là:

**C#**: Ngôn ngữ lập trình chính của .NET, được thiết kế để phát triển ứng dụng mạnh mẽ và dễ bảo trì.

**Visual Basic (VB.NET)**: Một ngôn ngữ lập trình hướng đối tượng dễ sử dụng.

**F#**: Ngôn ngữ lập trình hàm hỗ trợ phát triển các ứng dụng tính toán khoa học và dữ liệu.

**d. ASP.NET**

ASP.NET là một framework trong .NET dành cho phát triển các ứng dụng web. Phiên bản hiện đại **ASP.NET Core** hỗ trợ xây dựng các ứng dụng web đa nền tảng với hiệu suất cao, bao gồm các dịch vụ API web (RESTful API) và các ứng dụng web MVC.

**e. Entity Framework (EF)**

Entity Framework là một ORM (Object Relational Mapping) giúp lập trình viên tương tác với cơ sở dữ liệu một cách dễ dàng thông qua việc ánh xạ các đối tượng trong ngôn ngữ lập trình thành các bảng trong cơ sở dữ liệu. EF giúp đơn giản hóa việc truy vấn, thêm, sửa, xóa dữ liệu mà không cần viết SQL thủ công.

**f. Xamarin**

Xamarin là một thành phần trong .NET giúp phát triển ứng dụng di động đa nền tảng, tức là có thể viết một lần và chạy trên cả iOS và Android. Xamarin sử dụng C# và .NET để phát triển các ứng dụng di động mà không cần phải viết mã riêng cho từng nền tảng.

**Các phiên bản của .NET**

* **.NET Framework**: Được phát hành từ năm 2002, phiên bản này chủ yếu hỗ trợ phát triển các ứng dụng trên hệ điều hành Windows.
* **.NET Core**: Ra mắt năm 2016, đây là phiên bản đa nền tảng và mã nguồn mở của .NET, hỗ trợ Windows, Linux, và macOS.
* **.NET 5, 6, 7**: Đây là sự hợp nhất giữa .NET Framework và .NET Core thành một nền tảng duy nhất. Kể từ .NET 5, .NET hỗ trợ phát triển ứng dụng trên nhiều loại thiết bị và hệ điều hành hơn, đồng thời cải thiện hiệu suất và tính năng.

**Ưu điểm của nền tảng .NET**

* **Đa nền tảng**: Kể từ .NET Core, .NET đã trở thành một nền tảng phát triển đa nền tảng, cho phép ứng dụng chạy trên Windows, macOS và Linux.
* **Hiệu suất cao**: .NET có hiệu suất vượt trội nhờ tối ưu hóa bộ nhớ và quản lý luồng, đặc biệt với ASP.NET Core cho phát triển ứng dụng web nhanh và hiệu quả.
* **Mã nguồn mở**: .NET Core và các phiên bản tiếp theo của .NET đều là mã nguồn mở, cung cấp sự minh bạch và đóng góp từ cộng đồng lập trình viên.
* **Hỗ trợ ngôn ngữ đa dạng**: .NET hỗ trợ nhiều ngôn ngữ lập trình như C#, F#, Visual Basic, giúp lập trình viên có nhiều lựa chọn tùy theo nhu cầu dự án.
* **Thư viện phong phú**: .NET có một hệ thống thư viện phong phú và mạnh mẽ, giúp lập trình viên dễ dàng thực hiện các chức năng từ đơn giản đến phức tạp.
* **Bảo mật tốt**: Với các tính năng như quản lý người dùng, chứng thực, mã hóa và kiểm soát quyền truy cập, .NET cung cấp một nền tảng bảo mật mạnh mẽ cho các ứng dụng doanh nghiệp.

**Nhược điểm của .NET**

* **Phức tạp với người mới**: Với những người mới bắt đầu lập trình, .NET có thể gây khó khăn vì cấu trúc lớn và bao gồm nhiều thành phần.
* **Kích thước lớn**: Ứng dụng .NET thường có kích thước lớn hơn so với một số nền tảng khác, đặc biệt khi phát triển ứng dụng đa nền tảng hoặc tích hợp nhiều thư viện.
* **Phụ thuộc vào Microsoft**: Dù là mã nguồn mở, nhưng nền tảng .NET vẫn chịu sự quản lý chủ yếu từ Microsoft, và người dùng cần theo dõi các cập nhật từ Microsoft để đảm bảo ứng dụng luôn tương thích.

**Ứng dụng của .NET**

* **Phát triển ứng dụng doanh nghiệp**: .NET là nền tảng phổ biến trong phát triển các ứng dụng quản lý doanh nghiệp, hệ thống thông tin, và các dịch vụ backend mạnh mẽ.
* **Phát triển ứng dụng web**: ASP.NET Core là công cụ phát triển các trang web hiện đại và các dịch vụ web RESTful API hiệu quả.
* **Phát triển ứng dụng di động**: Với Xamarin, các lập trình viên có thể phát triển ứng dụng di động chạy trên iOS và Android từ cùng một mã nguồn C#.
* **Ứng dụng IoT và AI**: .NET cung cấp khả năng phát triển các ứng dụng thông minh với tích hợp trí tuệ nhân tạo và các thiết bị IoT.

## Giới thiệu về ngôn ngữ lập trình C#

**C# (C-Sharp)** là một ngôn ngữ lập trình hiện đại, hướng đối tượng và mạnh mẽ được phát triển bởi Microsoft, lần đầu tiên ra mắt vào năm 2000 như một phần của nền tảng .NET. Ngôn ngữ C# được thiết kế để kết hợp các yếu tố tốt nhất từ các ngôn ngữ lập trình phổ biến như C++, Java và Visual Basic, nhằm tạo ra một ngôn ngữ dễ sử dụng nhưng vẫn mạnh mẽ và hiệu quả trong việc phát triển các ứng dụng phần mềm. C# được sử dụng rộng rãi để phát triển các ứng dụng doanh nghiệp, ứng dụng web, trò chơi điện tử, ứng dụng di động và nhiều loại ứng dụng khác.

**Lịch sử phát triển**

C# được phát triển dưới sự chỉ đạo của **Anders Hejlsberg**, một lập trình viên nổi tiếng đã tham gia vào nhiều dự án phát triển ngôn ngữ lập trình quan trọng. Microsoft phát triển C# như một phần của sáng kiến .NET để cung cấp một công cụ phát triển linh hoạt, hiện đại và tích hợp tốt với các dịch vụ của Microsoft.

Các phiên bản C# sau này tiếp tục cải tiến với nhiều tính năng mới nhằm hỗ trợ lập trình viên phát triển ứng dụng một cách dễ dàng hơn. Tính đến nay, C# đã có nhiều phiên bản với các cải tiến mạnh mẽ về cú pháp, hiệu suất và khả năng tích hợp với các nền tảng đa dạng.

**Đặc điểm chính của C#**

C# kết hợp các tính năng của nhiều ngôn ngữ lập trình khác nhau, giúp nó trở nên linh hoạt và dễ học đối với nhiều lập trình viên. Các đặc điểm chính của C# bao gồm:

**a. Hướng đối tượng (Object-Oriented Programming - OOP)**

C# hỗ trợ đầy đủ lập trình hướng đối tượng, bao gồm các khái niệm như **kế thừa (inheritance)**, **đa hình (polymorphism)**, **đóng gói (encapsulation)**, và **trừu tượng hóa (abstraction)**. Điều này giúp tổ chức mã nguồn dễ hiểu, dễ bảo trì và mở rộng.

**b. Mạnh mẽ và an toàn**

C# được thiết kế để giảm thiểu các lỗi lập trình phổ biến và tăng cường tính an toàn. Ngôn ngữ cung cấp cơ chế quản lý bộ nhớ tự động qua **garbage collection**, giảm thiểu nguy cơ rò rỉ bộ nhớ. Ngoài ra, C# còn hỗ trợ kiểm tra kiểu tĩnh (static typing), giúp phát hiện lỗi sớm trong quá trình biên dịch.

**c. Hỗ trợ đa ngôn ngữ và đa nền tảng**

C# có thể chạy trên nhiều nền tảng nhờ .NET Core và các phiên bản sau này của .NET. Điều này cho phép các ứng dụng viết bằng C# có thể chạy trên **Windows, Linux, macOS** và nhiều thiết bị khác như **Android** và **iOS** thông qua **Xamarin**.

**d. Đơn giản và dễ học**

C# có cú pháp rõ ràng và dễ học, đặc biệt đối với những lập trình viên đã quen thuộc với các ngôn ngữ như Java hoặc C++. C# loại bỏ nhiều tính năng phức tạp của C++ như con trỏ (pointers) và quản lý bộ nhớ thủ công, giúp lập trình viên tập trung hơn vào logic ứng dụng.

**e. Được hỗ trợ bởi một thư viện phong phú**

C# có thể tận dụng **.NET Framework** và **.NET Core**, cung cấp hàng ngàn thư viện sẵn có để hỗ trợ phát triển các chức năng từ đơn giản đến phức tạp như truy cập dữ liệu, xử lý tệp, quản lý giao tiếp mạng, bảo mật, và nhiều tính năng khác.

**Các tính năng nổi bật của C#**

* **LINQ (Language Integrated Query)**: Giúp lập trình viên thực hiện các truy vấn dữ liệu ngay trong ngôn ngữ C#, hỗ trợ nhiều nguồn dữ liệu khác nhau như cơ sở dữ liệu SQL, bộ sưu tập trong bộ nhớ (in-memory collections), XML, v.v.
* **Asynchronous Programming (Lập trình bất đồng bộ)**: Hỗ trợ lập trình bất đồng bộ với từ khóa **async** và **await**, giúp tăng hiệu suất và khả năng phản hồi của ứng dụng mà không làm gián đoạn giao diện người dùng.
* **Lambda Expressions và Delegates**: Cung cấp cú pháp ngắn gọn cho các hàm vô danh (anonymous functions) và cách xử lý các phương thức như một đối tượng.
* **Event Handling**: Hỗ trợ mạnh mẽ cho lập trình sự kiện, đặc biệt hữu ích trong phát triển ứng dụng giao diện đồ họa và xử lý các sự kiện người dùng.
* **Generics**: Cho phép tạo ra các lớp và phương thức có thể làm việc với bất kỳ kiểu dữ liệu nào, mang lại sự linh hoạt và khả năng tái sử dụng mã cao.
* **Exception Handling**: C# có cơ chế xử lý ngoại lệ mạnh mẽ thông qua các khối lệnh **try**, **catch**, và **finally**, giúp quản lý lỗi một cách an toàn.

**Ứng dụng của C#**

C# là một ngôn ngữ đa năng, có thể sử dụng để phát triển nhiều loại ứng dụng khác nhau:

* **Phát triển ứng dụng doanh nghiệp**: C# thường được sử dụng để phát triển các ứng dụng doanh nghiệp lớn, đặc biệt là trên nền tảng .NET với các dịch vụ backend mạnh mẽ.
* **Phát triển ứng dụng web**: Với **ASP.NET Core**, C# được sử dụng rộng rãi để xây dựng các ứng dụng web hiện đại, dịch vụ web API, và các ứng dụng MVC (Model-View-Controller).
* **Phát triển ứng dụng di động**: C# được sử dụng trong **Xamarin** để phát triển các ứng dụng di động đa nền tảng chạy trên iOS và Android.
* **Phát triển trò chơi**: C# là ngôn ngữ chính được sử dụng trong **Unity**, một trong những nền tảng phát triển trò chơi phổ biến nhất trên thế giới.
* **Phát triển ứng dụng desktop**: C# được sử dụng để phát triển các ứng dụng desktop trên Windows thông qua **Windows Forms**, **WPF (Windows Presentation Foundation)**, và **UWP (Universal Windows Platform)**.
* **Ứng dụng IoT**: C# cũng có thể được sử dụng để phát triển các ứng dụng cho các thiết bị IoT, đặc biệt là khi kết hợp với Azure IoT của Microsoft.

**5. Ưu điểm của C#**

* **Hiệu suất cao**: C# kết hợp hiệu suất cao của ngôn ngữ biên dịch (compiled language) với sự tiện lợi của lập trình hướng đối tượng và hỗ trợ lập trình bất đồng bộ.
* **Dễ sử dụng và bảo trì**: C# có cú pháp gọn gàng, dễ hiểu, giúp viết mã dễ dàng hơn và bảo trì các dự án lớn hiệu quả hơn.
* **Tính nhất quán và linh hoạt**: C# có tính nhất quán cao trong cú pháp và các khái niệm lập trình, dễ học và dễ chuyển đổi từ các ngôn ngữ lập trình khác.
* **Bảo mật tốt**: C# có nhiều tính năng bảo mật tích hợp như kiểm tra kiểu tĩnh, ngăn chặn lỗi tràn bộ đệm (buffer overflow) và quản lý bộ nhớ tự động qua garbage collection.

**6. Nhược điểm của C#**

* **Chỉ tối ưu hóa trên hệ sinh thái Microsoft**: Mặc dù C# đã trở nên đa nền tảng với .NET Core, nhưng nó vẫn phát triển tối ưu nhất trong hệ sinh thái của Microsoft, và có thể không phải là lựa chọn tốt nhất cho các ứng dụng không chạy trên các nền tảng này.
* **Hiệu suất thấp hơn so với C++**: C# có hiệu suất thấp hơn một chút so với C++ khi xử lý các tác vụ yêu cầu hiệu suất cao, như các ứng dụng nhúng hoặc các trò chơi yêu cầu đồ họa phức tạp.
* **Khối lượng lớn**: Ứng dụng C# đôi khi yêu cầu nhiều tài nguyên hơn so với các ngôn ngữ khác, đặc biệt là khi chạy trên các nền tảng nhỏ gọn.

## Tổng quan về Framework ASP.NET MVC 5

**ASP.NET MVC 5** là một trong những framework mạnh mẽ và phổ biến được phát triển bởi **Microsoft**, nằm trong nền tảng **.NET**. Đây là phiên bản thứ năm của ASP.NET MVC (Model-View-Controller), cung cấp các công cụ phát triển ứng dụng web dựa trên mẫu kiến trúc **MVC**. Với việc phân tách rõ ràng giữa giao diện, logic và dữ liệu, ASP.NET MVC 5 giúp lập trình viên xây dựng các ứng dụng web dễ bảo trì, linh hoạt và mở rộng.

**Kiến trúc Model-View-Controller (MVC)**

MVC là một mẫu kiến trúc phần mềm, chia một ứng dụng thành ba phần chính: **Model**, **View**, và **Controller**. Điều này giúp tách biệt rõ ràng giữa giao diện người dùng, logic nghiệp vụ và dữ liệu.

* **Model**: Là thành phần đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng. Model chịu trách nhiệm quản lý dữ liệu, giao tiếp với cơ sở dữ liệu và xử lý logic nghiệp vụ.
* **View**: Là phần giao diện người dùng. View hiển thị thông tin cho người dùng dựa trên dữ liệu được cung cấp bởi Model. Trong ASP.NET MVC, View thường được viết bằng HTML kết hợp với Razor, một ngôn ngữ cú pháp giúp nhúng mã C# vào trong HTML.
* **Controller**: Là thành phần xử lý các yêu cầu từ người dùng, tương tác với Model để lấy dữ liệu và trả về View để hiển thị dữ liệu đó. Controller chịu trách nhiệm điều phối luồng dữ liệu giữa Model và View.

Kiến trúc MVC mang lại nhiều lợi ích như khả năng dễ dàng mở rộng, bảo trì, kiểm thử độc lập các thành phần, và tăng tính tái sử dụng mã.

**Tính năng nổi bật của ASP.NET MVC 5**

ASP.NET MVC 5 đi kèm với nhiều tính năng mạnh mẽ và hiện đại, hỗ trợ lập trình viên xây dựng các ứng dụng web hiệu quả và dễ bảo trì:

**a. Routing linh hoạt (Attribute Routing)**

Routing là cơ chế trong ASP.NET MVC để ánh xạ các URL tới các phương thức của Controller. Trong MVC 5, **attribute routing** cho phép lập trình viên định nghĩa các tuyến đường ngay trong các action của Controller thông qua việc sử dụng các thuộc tính (attribute). Điều này giúp tạo ra các URL thân thiện, dễ hiểu và có cấu trúc rõ ràng.

**b. Bảo mật mạnh mẽ (Authentication & Authorization)**

ASP.NET MVC 5 tích hợp sẵn **ASP.NET Identity**, một hệ thống xác thực và phân quyền người dùng hiện đại, cho phép lập trình viên dễ dàng quản lý tài khoản, phân quyền và kiểm soát truy cập. Hệ thống này hỗ trợ các phương pháp xác thực phổ biến như **OAuth**, **OpenID**, **Facebook**, **Google**, và **Twitter**.

**c. Bootstrap tích hợp**

Phiên bản MVC 5 hỗ trợ tích hợp với **Bootstrap**, một framework CSS phổ biến giúp xây dựng giao diện web hiện đại, phản hồi nhanh và thân thiện với người dùng. Việc tích hợp sẵn Bootstrap giúp lập trình viên dễ dàng tạo ra giao diện đẹp mắt mà không cần phải viết CSS từ đầu.

**d. Async Programming (Lập trình bất đồng bộ)**

MVC 5 hỗ trợ lập trình bất đồng bộ (async), cho phép xử lý các yêu cầu một cách không đồng bộ, giúp cải thiện hiệu suất ứng dụng, đặc biệt khi thực hiện các thao tác I/O chậm như truy vấn cơ sở dữ liệu hoặc gọi các dịch vụ web từ xa.

**e. Kết hợp với Web API**

ASP.NET MVC 5 có thể tích hợp với **ASP.NET Web API** để xây dựng các dịch vụ web RESTful. Điều này cho phép lập trình viên dễ dàng phát triển cả ứng dụng web và các dịch vụ API phục vụ các ứng dụng khác (như ứng dụng di động, ứng dụng bên thứ ba) chỉ trong một dự án duy nhất.

**f. Bundling & Minification**

Bundling (gói) và minification (tối ưu hóa) là hai tính năng giúp giảm thiểu số lượng và kích thước các tập tin CSS và JavaScript. Điều này giúp tối ưu hóa tốc độ tải trang web bằng cách giảm băng thông và thời gian tải của người dùng.

**g. Global Action Filters**

MVC 5 cung cấp **Action Filters** cho phép chặn các yêu cầu hoặc xử lý chúng trước và sau khi các action của Controller được thực thi. Lập trình viên có thể tạo ra các action filter tùy chỉnh để áp dụng các chức năng toàn cầu như logging, xử lý lỗi, hoặc kiểm tra quyền truy cập một cách tự động.

**Ưu điểm của ASP.NET MVC 5**

* **Tách biệt rõ ràng các thành phần**: Với kiến trúc MVC, việc phân tách Model, View và Controller giúp mã nguồn dễ bảo trì, mở rộng và kiểm thử.
* **Kiểm soát chi tiết quá trình phát triển giao diện**: ASP.NET MVC không sử dụng các **server controls** như trong Web Forms, điều này cho phép lập trình viên có quyền kiểm soát trực tiếp HTML và các tài nguyên giao diện, giúp tối ưu hóa tốt hơn cho SEO và các yếu tố hiệu suất.
* **Hỗ trợ tốt cho kiểm thử (Testability)**: MVC 5 cho phép kiểm thử unit (unit test) hiệu quả nhờ việc tách biệt các thành phần. Lập trình viên có thể dễ dàng kiểm thử từng phần riêng biệt mà không ảnh hưởng đến toàn bộ ứng dụng.
* **Hỗ trợ lập trình bất đồng bộ (Asynchronous Programming)**: Lập trình viên có thể sử dụng các phương thức async để tăng cường hiệu suất, đặc biệt là khi xử lý các yêu cầu I/O nặng như truy cập cơ sở dữ liệu hoặc API bên ngoài.
* **Cộng đồng lớn và hỗ trợ mạnh mẽ từ Microsoft**: Với sự hỗ trợ chính thức từ Microsoft và cộng đồng lập trình viên đông đảo, ASP.NET MVC có rất nhiều tài liệu, khóa học, và các công cụ hỗ trợ.

**Nhược điểm của ASP.NET MVC 5**

* **Đường cong học tập cao**: Đối với những lập trình viên mới hoặc đã quen với Web Forms, việc chuyển sang MVC có thể gặp khó khăn do MVC yêu cầu lập trình viên phải có hiểu biết sâu hơn về các khái niệm như routing, view engines, dependency injection, và HTTP request pipeline.
* **Không hỗ trợ các thành phần server controls**: Không giống như Web Forms, ASP.NET MVC không cung cấp các server controls, điều này có thể khiến cho việc phát triển giao diện cần phải viết mã HTML, CSS và JavaScript nhiều hơn.
* **Tùy chỉnh nhiều**: Do ASP.NET MVC không dựa vào các thành phần giao diện có sẵn như Web Forms, lập trình viên phải tự viết và tùy chỉnh rất nhiều về giao diện, điều này có thể làm tăng thời gian phát triển ban đầu.

# **CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI**

## Đặt vấn đề

Trong thời đại công nghệ số phát triển mạnh mẽ, việc mua sắm trực tuyến ngày càng trở nên phổ biến. Nhiều ngành hàng, bao gồm thời trang, mỹ phẩm và đặc biệt là nước hoa, đã chuyển dịch từ mô hình kinh doanh truyền thống sang kinh doanh online. Tuy nhiên, việc xây dựng và phát triển một website bán nước hoa không đơn giản chỉ là đưa sản phẩm lên mạng và chờ đợi khách hàng đến mua. Nó đòi hỏi sự đầu tư kỹ lưỡng về mặt chiến lược, trải nghiệm người dùng, bảo mật cũng như uy tín của doanh nghiệp.

Nước hoa là một mặt hàng cao cấp, liên quan đến cảm xúc và sở thích cá nhân, do đó, khách hàng thường có yêu cầu cao về cả chất lượng sản phẩm và dịch vụ. Một trang web bán nước hoa không chỉ cần cung cấp thông tin đầy đủ, hình ảnh đẹp mắt, mà còn phải xây dựng được lòng tin nơi người tiêu dùng. Sự minh bạch về nguồn gốc sản phẩm, chính sách đổi trả và chăm sóc khách hàng tốt là những yếu tố cần thiết để tạo dựng niềm tin, đặc biệt khi khách hàng không thể trực tiếp thử mùi hương.

Ngoài ra với sự cạnh tranh gay gắt trên thị trường thương mai điện tử , việc tối ưu hóa trải nghiệm của người dùng là vô cùng quan trọng.Một website bắt mắt ,dễ sử dụng sẽ giúp người mua dễ dàng tiếp cận với những sản phẩm phù hợp với nhu cầu của họ . Không chỉ vậy, các tính năng thanh toán an toàn và tiện lợi, chính sách vận chuyển rõ ràng cũng góp phần tạo nên sự thành công của một website bán nước hoa.

Như vậy, để phát triển một trang web bán nước hoa trực tuyến hiệu quả, cần giải quyết một loạt các vấn đề quan trọng từ việc nghiên cứu thị trường, thiết kế website, xây dựng niềm tin, đến chiến lược tiếp thị và dịch vụ khách hàng. Đây là những yếu tố cơ bản nhưng vô cùng thiết yếu để có thể cạnh tranh và phát triển bền vững trong lĩnh vực kinh doanh nước hoa trực tuyến.

## Hệ thống hiện tại

* **Giao diện người dùng(UI/UX):** Phầ hiển thị mà khách hàng nhìn thấy khi truy cập vào trang web.Bao gồm giao diện thiết kế bắt mắt.màu sắc hài hòa,cấu trúc dễ sử dụng.
* **Quản lý sản phẩm**: Hệ thống sản phẩm giúp quản trị viên theo dõi và cập nhật thông tin về các sản phẩm nước hoa như tên,mô tả, hình ảnh.Hệ thống này tích hợp các công cụ giúp dễ dàng thêm,xóa hoặc chỉnh sửa sản phẩm.
* **Giỏ hàng và thanh toán**: Hệ thống giỏ hàng giúp người dùng lưu trữ các sản phẩm đã chọn để mua sau đó .Phần thanh toán cần tích hợp các phương thức thanh toán phổ biến như thẻ tín dụng, ví điện tử, và chuyển khoản ngân hàng. Đảm bảo bảo mật thông tin thanh toán và mã hóa dữ liệu để ngăn chặn rủi ro từ các cuộc tấn công mạng là điều cần thiết.
* **Quản lý hệ đơn hàng và vận chuyển**: Hệ thống theo dõi đơn hàng từ lúc khách hàng đặt hàng đến lúc nhận Nó bao gồm tính năng quản lý trạng thái đơn hàng , thông tin vận chuyển, thông báo thời gian giao hàng.Liên kết đối tác vận chuyển để tự động cập nhật quá trình giao hàng.
* **Marketing và quảng bá**: Hệ thống tiếp thị thích hợp các công cụ SEO để tối ưu hóa thứ hạng tìm kiếm của trang web.Điều này giúp tiếp cận và thu hút khách hàng mới cũng như duy trì sự trung thành của khách hàng hiện tại.
* **Chăm sóc khách hàng**: Hệ thống quản lý và tương tác với khách hàng qua các kênh như chat trực tuyến,email,hotline.Điều này giúp nhanh chóng giải quyết các vấn đề phát sinh trong quá trình mua hàng hoặc tư vấn sản phẩm

## Hệ thống đề xuất

* **Hệ thống quản lý sản phẩm thông minh**:

+ **Cở sở dữ liệu sản phẩm chi tiết**: Mỗi sản phẩm nên có thông tin mô tả chi tiết về thương hiệu, loại nước hoa (hương đầu, hương giữa, hương cuối), giới tính, giá cả, dung tích, và hình ảnh chất lượng cao.

+**Quản lý tồn kho tự động**: Tích hợp tính năng tự động cập nhật số lượng tồn kho, cảnh báo khi sắp hết hàng, và đề xuất sản phẩm thay thế.

+**Công cụ phân loại sản phẩm thông minh**: Phân loại theo thương hiệu, nhóm hương (hoa, gỗ, trái cây), giới tính, giá cả, dung tích. Đồng thời, cần có công cụ gợi ý sản phẩm dựa trên sở thích và lịch sử mua hàng của khách hàng.

* **Hệ thống tìm kiếm và gợi ý cá nhân hóa:**

**+ Công cụ tìm kiếm nâng cao:** Tìm kiếm theo từ khóa, lọc theo các thuộc tính sản phẩm (thương hiệu, nhóm hương, giá cả).

**+ Gợi ý sản phẩm theo hành vi mua hàng:** Sử dụng trí tuệ nhân tạo (AI) để phân tích hành vi người dùng, sau đó gợi ý các sản phẩm tương tự hoặc phù hợp với sở thích cá nhân của họ.

**+ Tính năng "Yêu thích":** Cho phép người dùng lưu lại các sản phẩm yêu thích để mua sau hoặc nhận thông báo khi có giảm giá.

* Hệ thống thanh toán và bảo mật:

+**Tích hợp nhiều phương thức thanh toán**: Hỗ trợ thanh toán qua thẻ tín dụng, ví điện tử (Momo, ZaloPay), chuyển khoản ngân hàng, hoặc thanh toán khi nhận hàng (COD).

+**Xác minh giao dịch và bảo mật SSL**: Mã hóa SSL cho toàn bộ trang web và các giao dịch thanh toán để bảo mật thông tin khách hàng. Tích hợp hệ thống xác minh đa lớp (OTP) để đảm bảo an toàn cho người mua.

+**Tự động tạo hóa đơn điện tử**: Cung cấp hóa đơn điện tử tự động qua email ngay khi thanh toán thành công.

* **Hệ thống quản lý đơn hàng và vận chuyển:**

**+ Theo dõi đơn hàng trực tuyến**: Cung cấp cho khách hàng công cụ theo dõi đơn hàng theo thời gian thực, từ lúc đặt hàng đến khi giao nhận**.**

**+Tích hợp với các đối tác vận chuyển:** Kết nối với các nhà vận chuyển uy tín để tự động cập nhật trạng thái giao hàng, đồng thời cho phép khách hàng chọn phương thức vận chuyển phù hợp (giao nhanh, giao thường).

**+Thông báo trạng thái qua SMS và email:** Gửi thông báo qua SMS và email khi có thay đổi trạng thái đơn hàng hoặc thông tin về giao hàng.

* **Hệ thống chăm sóc khách hàng và hỗ trợ trực tuyến:**

**+Tích hợp chat bot và hỗ trợ trực tiếp**: Sử dụng chatbot hỗ trợ tự động để trả lời các câu hỏi thông thường, kèm theo hệ thống hỗ trợ trực tiếp (live chat) để giải quyết các vấn đề phức tạp.

**+Hệ thống phản hồi và đánh giá sản phẩm:** Khuyến khích khách hàng để lại đánh giá về sản phẩm, qua đó tạo niềm tin và giúp cải thiện chất lượng dịch vụ.

**+Trung tâm trợ giúp và FAQ:** Cung cấp phần FAQ (câu hỏi thường gặp) với hướng dẫn chi tiết về mua sắm, thanh toán và bảo hành, giúp khách hàng dễ dàng tìm hiểu thông tin trước khi liên hệ hỗ trợ.

* **Hệ thống tiếp thị và quảng bá:**

**+Tích hợp công cụ SEO:** Tối ưu hóa trang web để tăng khả năng xuất hiện trên các công cụ tìm kiếm như Google, giúp thu hút khách hàng tiềm năng.

**+Tự động gửi email marketing:** Hệ thống tự động gửi email thông báo về các chương trình khuyến mãi, sản phẩm mới hoặc nhắc nhở khách hàng về giỏ hàng bị bỏ quên.

**+Tính năng tích điểm khách hàng:** Tạo các chương trình tích điểm hoặc ưu đãi thành viên để khuyến khích khách hàng quay lại mua sắm.

* **Hệ thống phân tích và báo cáo:**

**+Theo dõi hành vi khách hàng**: Sử dụng công cụ phân tích dữ liệu để theo dõi hành vi mua sắm, từ đó tối ưu hóa trải nghiệm người dùng và chiến lược tiếp thị.

**+Báo cáo doanh số và hiệu quả quảng cáo**: Cung cấp báo cáo chi tiết về doanh số, đơn hàng, và hiệu quả của các chiến dịch quảng cáo, giúp quản trị viên đưa ra các quyết định kinh doanh chính xác.

**+Hệ thống đo lường tỷ lệ chuyển đổi**: Theo dõi tỷ lệ khách truy cập trang web và khách hàng thực sự mua hàng, từ đó điều chỉnh chiến lược kinh doanh và cải thiện hiệu suất bán hàng.

* **Tích hợp ứng dụng di động:**

**+Ứng dụng di động dành cho khách hàng**: Phát triển ứng dụng di động cho cả Android và iOS, giúp khách hàng dễ dàng truy cập, mua sắm mọi lúc, mọi nơi với giao diện thân thiện trên điện thoại.

**+Thông báo đẩy (Push Notification)**: Gửi thông báo về các ưu đãi, chương trình giảm giá hoặc sản phẩm mới qua ứng dụng di động để thu hút sự chú ý của khách hàng.

## Giới hạn của hệ thống

* **Giới hạn về trải nghiệm cảm giác thực tế (mùi hương)**

**+Không thể thử mùi hương**: Khách hàng không thể trực tiếp trải nghiệm mùi hương nước hoa khi mua hàng trực tuyến, điều này có thể khiến họ khó quyết định chọn mua sản phẩm, đặc biệt với những loại nước hoa mới hoặc chưa được biết đến.

**+Hình ảnh và mô tả không thay thế được cảm nhận thực tế**: Dù hệ thống có thể cung cấp hình ảnh và mô tả chi tiết, nhưng chúng không thể thay thế trải nghiệm trực tiếp khi thử nước hoa tại cửa hàng.

* **Khả năng gợi ý sản phẩm phụ thuộc vào dữ liệu và thuật toán**

**+Dữ liệu khách hàng hạn chế**: Nếu hệ thống không thu thập đủ thông tin từ khách hàng hoặc hành vi của họ, tính năng gợi ý sản phẩm cá nhân hóa có thể kém chính xác hoặc không hữu ích.

**+Chất lượng thuật toán gợi ý**: Nếu thuật toán AI không đủ mạnh hoặc không được tối ưu hóa, hệ thống có thể gợi ý những sản phẩm không phù hợp với sở thích của khách hàng, từ đó làm giảm trải nghiệm mua sắm.

* **Bảo mật và rủi ro thanh toán**

**+Rủi ro tấn công mạng**: Mặc dù hệ thống đã tích hợp các công cụ bảo mật như SSL và mã hóa giao dịch, nhưng các cuộc tấn công mạng, lừa đảo hoặc đánh cắp dữ liệu vẫn có thể xảy ra, đặc biệt nếu hệ thống không được cập nhật thường xuyên và bảo trì tốt.

**+Sự tin tưởng của khách hàng**: Một số khách hàng vẫn ngại thanh toán trực tuyến do lo ngại về an toàn và bảo mật, đặc biệt trong bối cảnh có nhiều vụ rò rỉ thông tin hoặc lừa đảo trực tuyến.

* **Khả năng xử lý đơn hàng và quản lý tồn kho**

**+Đồng bộ với kho hàng thực tế**: Việc cập nhật số lượng hàng tồn kho có thể gặp vấn đề nếu hệ thống không được đồng bộ hóa tốt với kho hàng thực tế. Điều này có thể dẫn đến việc bán quá số lượng hàng có sẵn hoặc không thể cung cấp sản phẩm kịp thời.

**+Quá tải trong thời gian cao điểm**: Trong các dịp khuyến mãi hoặc ngày lễ, hệ thống quản lý đơn hàng có thể gặp khó khăn trong việc xử lý khối lượng đơn hàng lớn cùng lúc, dẫn đến chậm trễ trong vận chuyển và làm giảm trải nghiệm khách hàng.

* **Phụ thuộc vào đối tác vận chuyển**

**+Chất lượng dịch vụ vận chuyển**: Trang web có thể gặp giới hạn trong việc kiểm soát dịch vụ vận chuyển, đặc biệt nếu đối tác vận chuyển chậm trễ, làm hỏng hàng hóa, hoặc không cung cấp dịch vụ tốt. Điều này có thể ảnh hưởng tiêu cực đến hình ảnh của trang web.

**+Phí vận chuyển cao**: Một số khu vực có thể yêu cầu phí vận chuyển cao hoặc dịch vụ vận chuyển không đáng tin cậy, dẫn đến việc hạn chế khả năng tiếp cận của khách hàng ở các khu vực xa.

* **Chi phí triển khai và duy trì hệ thống**

**+Chi phí phát triển ban đầu cao**: Để xây dựng một hệ thống hoàn chỉnh và hiện đại, doanh nghiệp sẽ phải đầu tư chi phí lớn cho phát triển website, ứng dụng di động, và các tính năng phức tạp như trí tuệ nhân tạo, bảo mật và quản lý tồn kho tự động.

**+Chi phí bảo trì và cập nhật**: Sau khi triển khai, việc duy trì và cập nhật hệ thống (ví dụ: vá lỗ hổng bảo mật, nâng cấp tính năng) cũng tốn kém. Nếu không được duy trì đều đặn, hệ thống có thể trở nên lỗi thời và gặp rủi ro về bảo mật.

* **Thách thức về tiếp thị trực tuyến**

**+Cạnh tranh mạnh mẽ**: Trong thị trường nước hoa trực tuyến, có nhiều đối thủ cạnh tranh với các chiến lược tiếp thị mạnh mẽ. Việc thu hút khách hàng tiềm năng có thể gặp khó khăn, đặc biệt nếu chi phí quảng cáo ngày càng cao.

**+Tỷ lệ chuyển đổi thấp**: Dù hệ thống có thu hút lượng lớn khách hàng truy cập, nhưng việc chuyển đổi từ người truy cập thành người mua thực sự có thể không cao, đặc biệt nếu khách hàng còn do dự về sản phẩm hoặc giá cả.

* **Hạn chế về tùy chỉnh và cá nhân hóa**

**+Giới hạn khả năng cá nhân hóa sâu**: Hệ thống có thể bị giới hạn trong việc cá nhân hóa sâu các dịch vụ và sản phẩm cho từng khách hàng, đặc biệt nếu không có đủ dữ liệu hoặc thuật toán chưa phát triển đến mức tối ưu hóa cho mọi khách hàng cá nhân.

**+Khó đoán sở thích khách hàng**: Với sản phẩm nước hoa, sở thích rất mang tính cá nhân, và có thể hệ thống không đủ chính xác để gợi ý sản phẩm hoàn toàn đúng theo mong muốn của người dùng.

## Yêu cầu về phần cứng và phần mềm

## Yêu cầu tối thiểu

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram: 4GB  \* HDD >=500MB free  \* CPU: …  \* Connect the Internet. | \* Ram: 2GB  \* CPU : 2 Core 4 Threads  \* Connect the Internet. |
| Software | \* Hệ điều hành: ….  \* SQL Server …  \* NetFramework:… | \* Hệ điều hành: …  \* 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:4GB  \* CPU: 2 Core 4  \*Connect the Internet. |
| Software | \* Hệ điều hành: Windows Server …..  \* SQL Server ….  \* Net Framework:…. | \* 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 nước hoa như tên nước hoa , kích cỡ, mùi hương, giá cả… |
| **Nhập** | Lựa chọn loại nước hoa cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* **Tìm kiếm sản phẩm**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể tìm kiếm thông tin về sản phẩm, dịch vụ hoặc thông tin liên quan bằng công cụ tìm kiếm trên trang. |
| **Nhập** | Lựa chọn loại nước hoa cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* **Thêm giỏ hàng**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể thêm sản phẩm vào giỏ hàng , nhưng để thực hiện các hành động như thanh toán hay lưu trữ lâu dài, họ có thể cần phải đăng nhập. |
| **Nhập** | Lựa chọn loại nước hoa cần thêm vào giỏ hàng. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* Try cập nội dung công khai

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể các bài viết, tin tức, hoặc thông tin công khai mà không cần phải đăng nhập, nhưng có thể bị hạn chế về số lượng nội dung. |
| **Nhập** | Lựa chọn loại nước hoa cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* **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**

|  |  |
| --- | --- |
| **Mô tả** | Thành viên có thể thêm sản phẩm vào giỏ hàng và đặt sản phẩm |
| **Nhập** | Nhập thông tin đặt hàng vào form đặt hàng |
| **Xử lý** | * Kiểm tra thông tin đặt hàng của khách hàng * Thêm và cập nhật vào csdl |
| **Xuất** | * Hiển thị chức năng cập nhật và xóa sản phẩm * Thông báo cho khách hàng đã hoàn tất đơn đặt hàng |

* + 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 sản phẩm**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc xóa ,chỉnh sửa , sắp xếp các danh mục các sản phẩm để người dùng mua được rõ ràng. |
| **Nhập** | Thêm: |
| **Xử lý** | * Thêm sản phẩm cho trang web   Sửa:   * Sửa các sản phẩm có sẵn trong trang web   Sắp xếp :   * Sắp xếp các danh mục sản phẩm rõ ràng hơn |
| **Xuất** | Hiện thông tin đã thay đổi. |

* **Quản lý đơn hàng**

|  |  |
| --- | --- |
| **Mô tả** | * Quản trị viên có thể xem được các sản phẩm khách hàng đã đặt |
| **Nhập** | * Nhập thông tin đơn vị vận chuyển đơn hàng |
| **Xử lý** | * Xác nhận trạng thái đơn hàng |
| **Xuất** | * Hoàn tất đơn hàng |

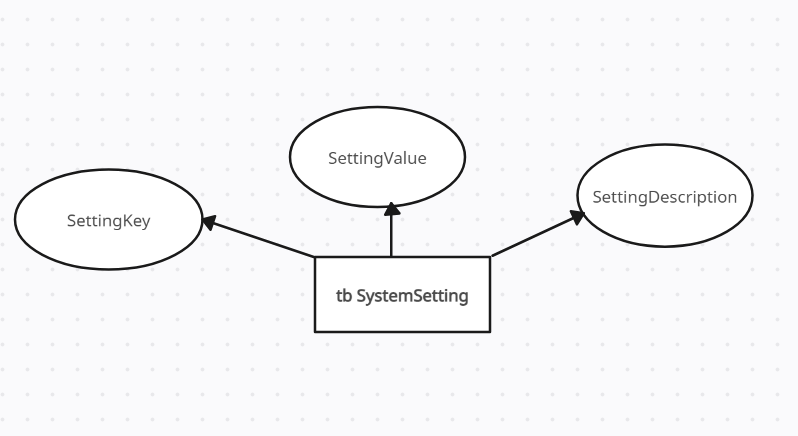
## Systeme Designs (Thiết kế hệ thống)

## Entity Relationship Diagram (Mô hình quan hệ thực thể)

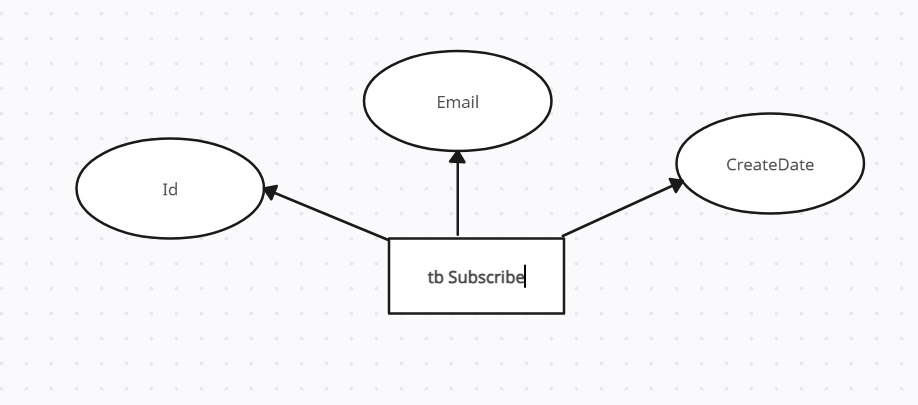
<< Trình bày nội dung mô hình ở đây >>

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

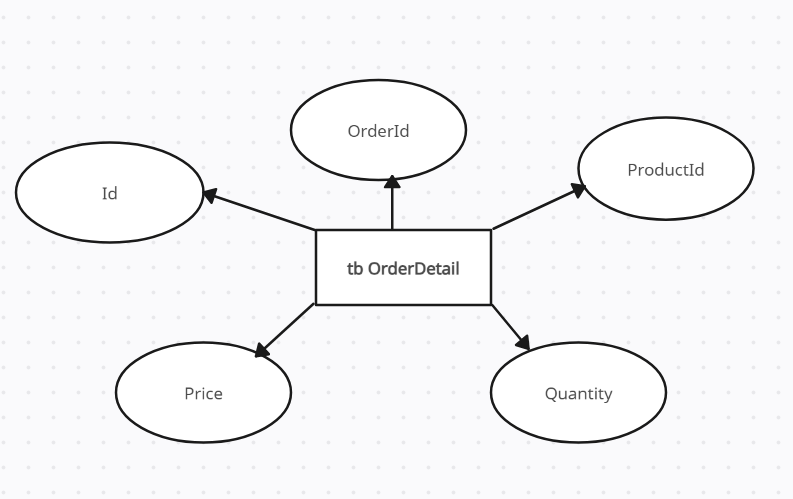
1. Thực thể thiết lập hệ thống (**tb SystemSetting**):   
   Lưu trữ thông tin người quản trị viên hệ thống.



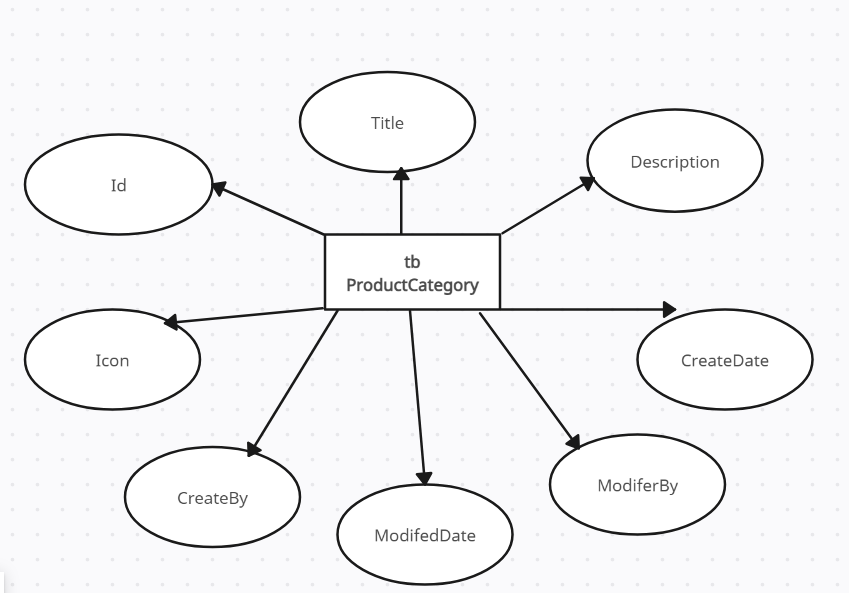
1. Thực thể đăng ký ( **tb Subscribe** ): Lưu trữ thông tin khách hàng khi đăng ký, mua hàng tại hệ thống.



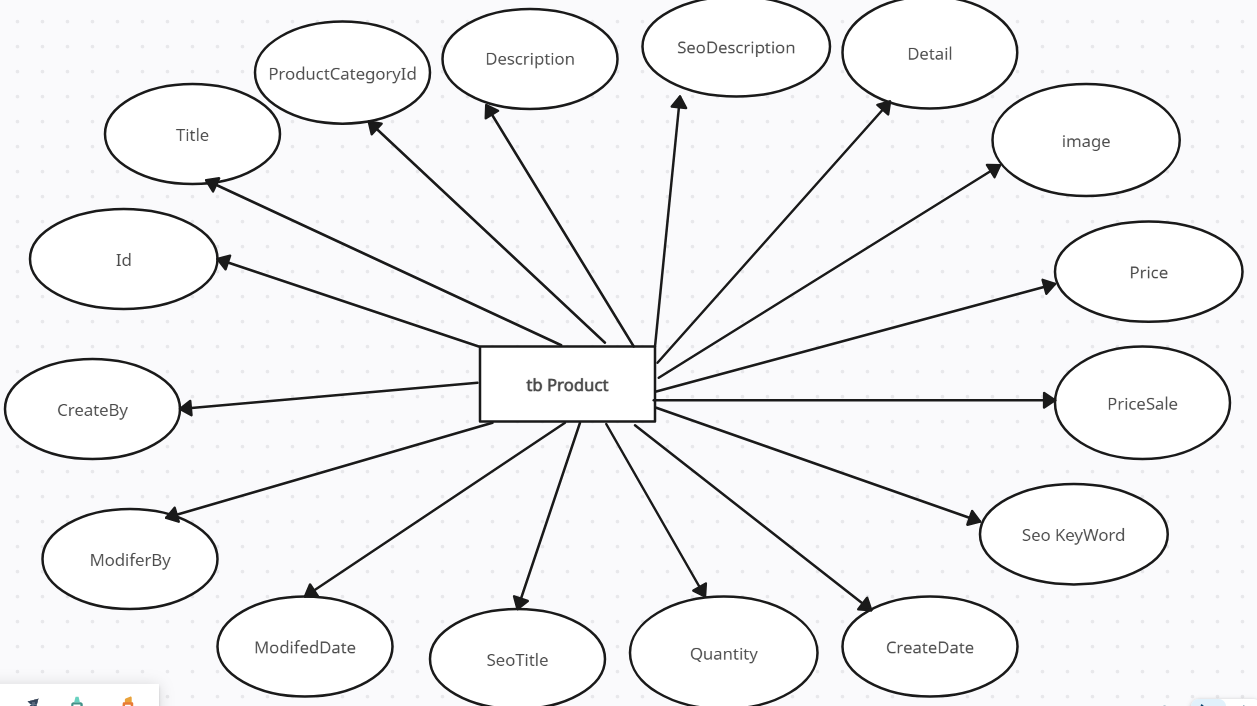
1. Thực thể chi tiết đơn hàng (**tb OrderDetail**): Lưu trữ thông tin đơn hàng khi khách hàng mua hàng tại hệ thống



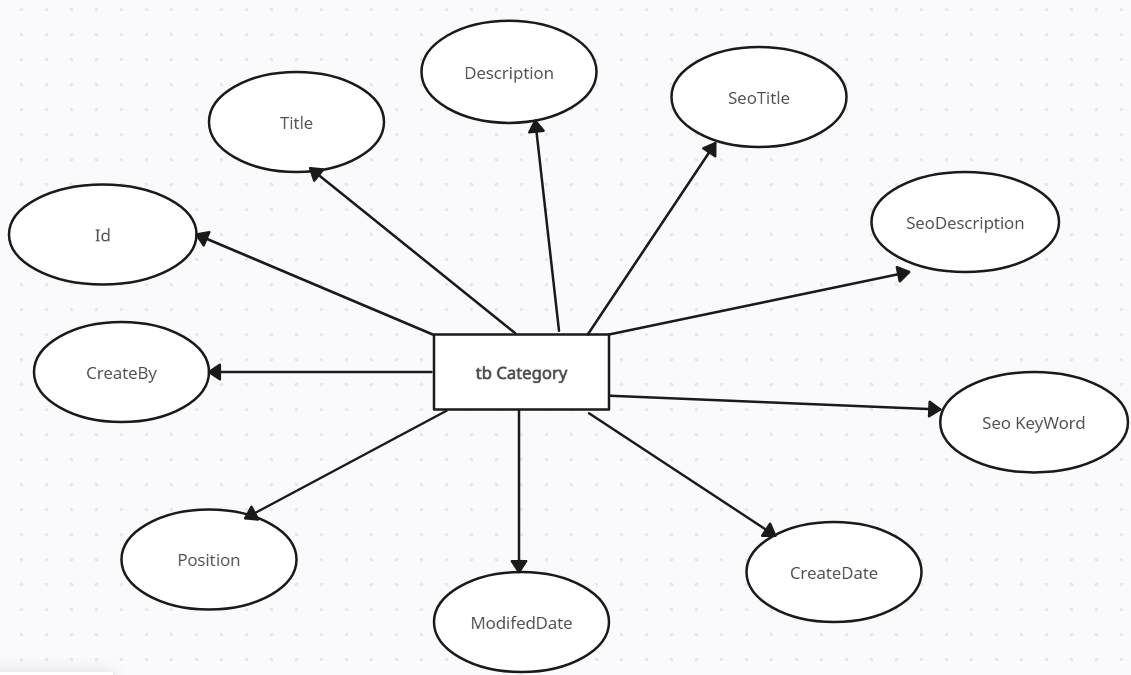
1. Thực hệ danh mục sản (**tb ProductCategory**): Lưu trữ thông tin danh mục sản phẩm của hệ thống.



1. Thực thể sản phẩm (**tb Product**): Lưu trữ thông tin sản phẩm của hệ thống



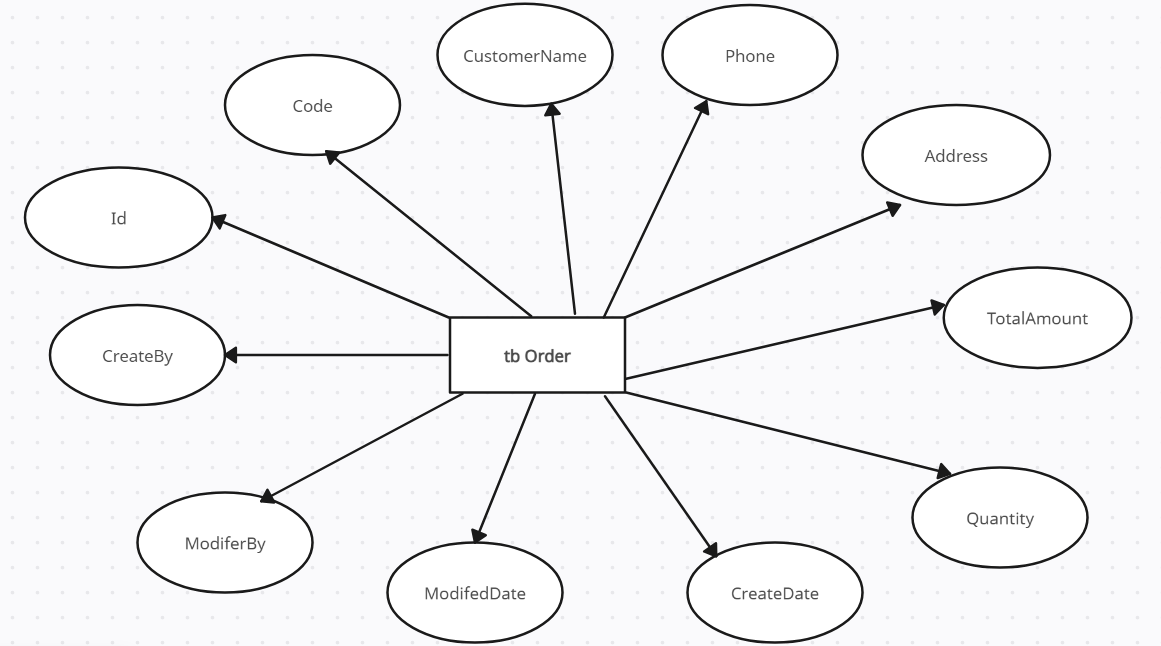
1. Thực thể danh mục (**tb Category**): Lưu trữ thông tin danh mục của hệ thống



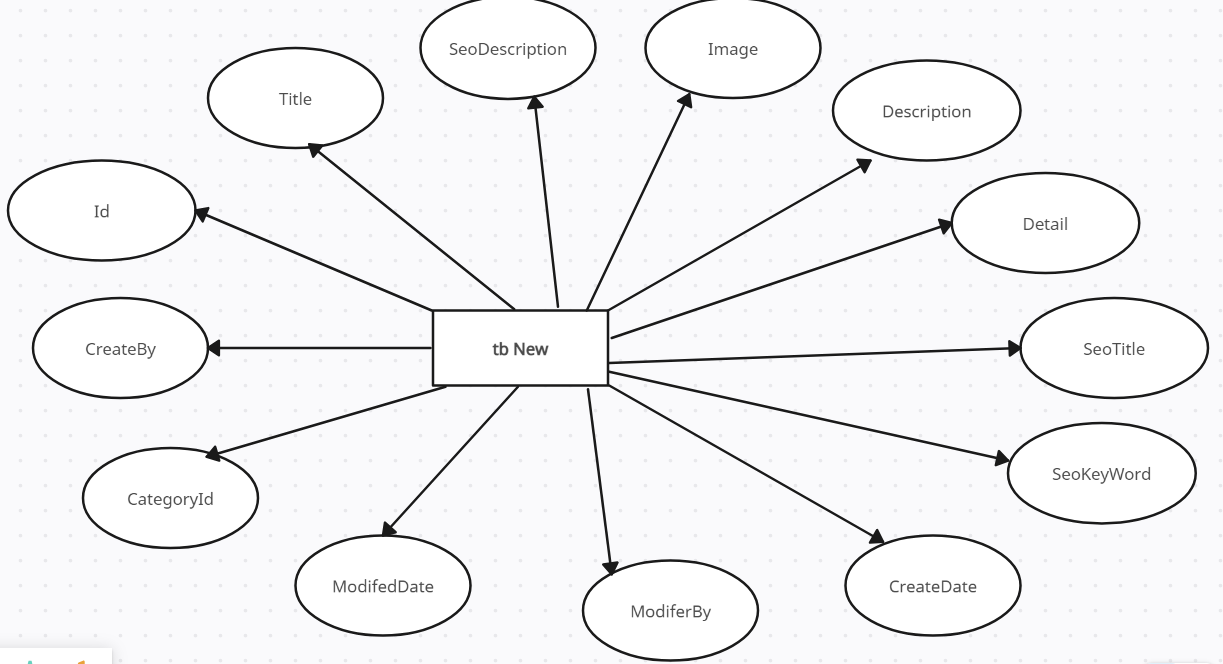
1. Thực thể bài viết(**tb Post**): Lưu trữ thông tin bài viết của hệ thống



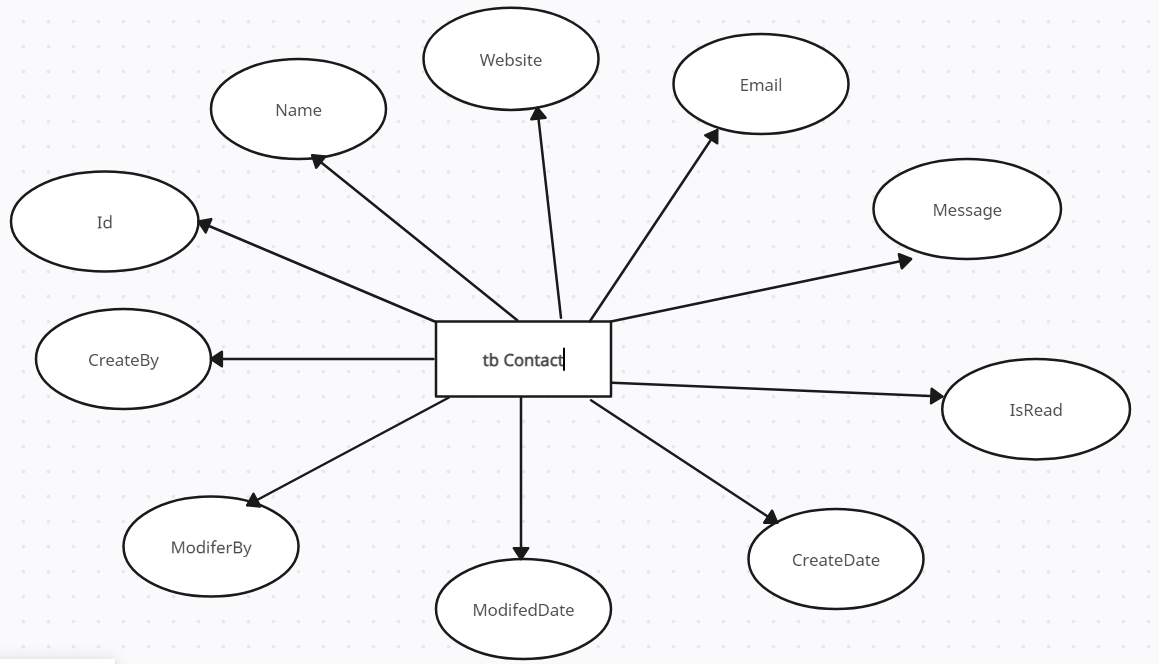
1. Thực thể đặt hàng(**tb Order**): Lưu trữ thông tin đặt hàng của khách hàng tại hệ thống



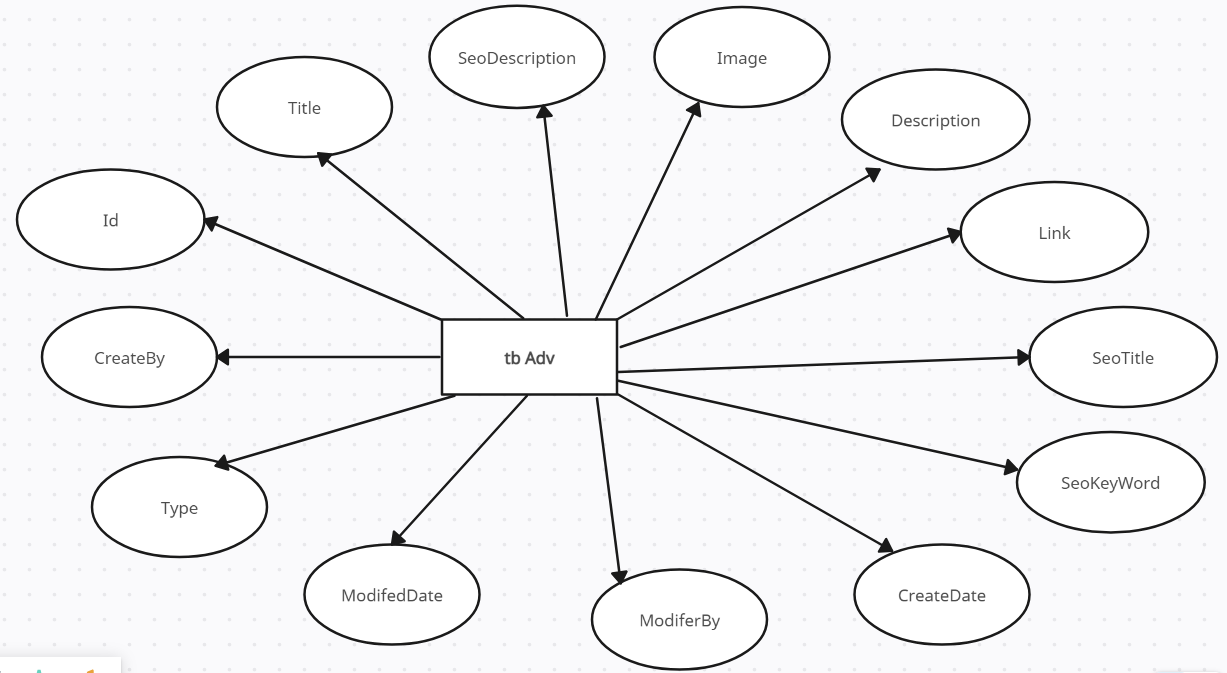
1. Thực thể sửa đổi (**tb New**) : Lưu trữ thông sửa đổi sản phẩm của hệ thống



1. Thực thể liên hệ(**tb Contact**) : Lưu trữ thông tin liên hệ của khách hàng tại hệ thống



1. Thực thể quảng cáo (**tb Adv**): Lưu trữ thông tin quảng cáo của 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 **tb OrderDetai**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**  **Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc**  **Constraint** | **Mô tả Content** |
| Id | int |  | Primary Key | Id |
| OrderId | int |  |  | Id đặt hàng |
| ProductId | int |  |  | Id sản phẩm |
| Price | Decimal(18,0) |  |  | Giá sản phẩm |
| Quantity | int |  |  | Số lượng sản phẩm |

1. Bảng **tb ProductCategory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc Constraint** | **Mô tả Content** |
| Id | int |  | Primary Key | Id |
| Title | Nvachar(150) |  |  | Tên danh mục sản phẩm |
| Description | Nvachar(500) |  |  | Mô tả danh mục sản phẩm |
| CreateDate | Datetime |  |  | Ngày tạo |
| ModififerBy | Nvarchar**(**150**)** |  |  | Sửa đổi bởi |
| ModifedDate | Datetime |  |  | Ngày sửa đổi |
| CreateBy | Nvarchar(150) |  |  | Tạo bởi |

1. Bảng tb Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc Constraint** | **Mô tả Content** |
| Id | int |  | Primary Key | Id Sản phẩm |
| Title | Nvarchar(250) |  |  | Tên sản phẩm |
| ProductCategoryID | int |  | Primary Key | Id danh mục sản phẩm |
| Description | Nvarchar(4000) |  |  | Mô tả |
| Detail | Nvarchar(MAX) |  |  | Chi tiết sản phẩm |
| Image | Nvarchar(500) |  |  | Ảnh sản phẩm |
| Price | Decimal(18,2) |  |  | Giá sản phẩm |
| PriceSale | Decimal(18,2) |  |  | Giá sản phẩm sale |
| Quantity | int |  |  | Số lượng sản phẩm |
| SeoTitle | Nvarchar(250) |  |  | Thẻ tiêu đề sản phẩm |
| SeoDescription | Nvarchar(550) |  |  | Tiêu đề trang sản phẩm |
| SeoKeyword | Nvarchar(250) |  |  | Từ khóa |
| CreateDate | datetime |  |  | Ngày tạo |
| CreateBy | Nvarchar(150) |  |  | Tạo bởi |
| ModifierBy | Nvarchar(150) |  |  |  |

1. Bảng tb Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc Constraint** | **Mô tả Content** |
| Id | int |  | Primary Key | Id loại sản phẩm |
| Title | Nvarchar(150) |  |  | Tên loại sản phẩm |
| Description | Nvarchar(500) |  |  | Mô tả loại sản phẩm |
| SeoTitle | Nvarchar(150) |  |  | Thẻ tiêu đề sản phẩm |
| SeoDescription | Nvarchar(550) |  |  | Tiêu đề trang sản phẩm |
| SeoKeyword | Nvarchar(250) |  |  | Từ khóa |
| Position | int |  |  | Vị trí |
| CreateDate | datetime |  |  | Ngày tạo |
| CreateBy | Nvarchar(150) |  |  | Tạo bởi |
| ModifedDate | datetime |  |  | Dùng để ghi chép lại tình trạng của tập tin hay thư mục |
| ModiferBy | Nvarchar(150) |  |  |  |

1. Bảng tb Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột Field** | **Kiểu dữ liệu DataType** | **Null** | **Ràng buộc Constraint** | **Mô tả Content** |
| Id | int |  | Primary Key | Id |
| Code | Nvarchar(50) |  |  |  |
| CustomerName | Nvarchar(150) |  |  | Tên khách hàng |
| Phone | Nvarchar(15) |  |  | Số điện thoại |
| Address | Nvarchar(500) |  |  | Địa chỉ |
| TotalAmount | Decimal(18,0) |  |  | Tổng số tiền |
| Quantity | int |  |  | Số lượng |
| CreateDate | Datetime |  |  | Ngày tạo |
| CreateBy | Nvarchar(150) |  |  | Tạo bởi |
| ModifedDate | datetime |  |  | Dùng để ghi chép lại tình trạng của tập tin hay thư mục |
| ModiferBy | Nvarchar(150) |  |  |  |

## Table RelationShip Diagram (Mô hình quan hệ bảng)

## SiteMap (Sơ đồ cấu trúc site)

<< nội dung sitemap thay đổi cho phù hợp với đề tài>>

## 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

## Algorithms (Giải thuật)

<< Bổ dung chỉnh sửa cho phù hợp với đề tài>>

## Đăng ký



|  |  |
| --- | --- |
| * **Kiểm tra thông tin nhập**:   Kiểm tra các thông tin bắt buộc nhập theo ràng buộc khi thiết kế cơ sở dữ liệu:   * + Tài khoản, Mật khẩu, Email, Điện thoại,.. | * **Kiểm tra sự tồn tại của tài khoản:**   Tài khoản là duy nhất đối với mỗi khách hàng. Nếu đã có người dùng rồi thì không cho phép đăng ký. |

## Đăng nhập



## Đăng xuất



## Đổi mật khẩu



## Thay đổi thông tin cá nhân



## Thêm sản phẩm



## Xóa sản phẩm



## Đặt mua



## Tìm kiếm



# **CHƯƠNG 4: XÂY DỰNG HỆ THỐNG**

## Xây dựng phần mềm

## Một số mã nguồn chương trình

<< Các mã nguồn cơ bản copy code vào đây>>

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

## Giao diện trang chủ

## Giao diện trang quản trị

## …..

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## Kết luận

Quá trình làm đồ án chuyên đề với đề tài “Trang web nư" 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:

**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 …..

## Task Sheet (bảng kế hoạch công việc)

|  |  |  |  |
| --- | --- | --- | --- |
| **Giai đoạn** | **Từ ngày 15/10 đến 25/10** | **Công việc** | **Tỉ lệ hoàn thành** |
| Chương 1:  (100%) | 15/10 | **Nghiên cứu cơ sở lí thuyết:**   * Tổng quan về lập trình thiết kế giao diện website * Tổng quan về hệ quản trị cơ sở dữ liệu SQL Sever * Tổng quan về ngôn ngữ lập trình phía máy chủ | 100% |
| Chương 2:  (100%) | 18/10-21/10 | **Tổng quan về đề tài:**   * Đặt vấn đề * Hệ thống hiện tại * Hệ thống đề xuất * Giới hạn hệ thống * Yêu cầu về phần cứng và phần mềm * Yêu cầu tối thiểu * Yêu cầu đề nghị | 100% |
| Chương 3:  (100%) | 22/10-24/10 | **Phân tích yêu cầu khách hàng và thiết kế hệ thống:**   * Người dùng của hệ thống * Chức năng của hệ thống * Systeme Design(Thiết kế hệ thống) * Database Design(Thiết kế cơ sở dữ liệu) * Table Relationship Diagram(Mô hình quan hệ bảng) * Site Map(Sơ đồ cấu trúc site) * Algorithms(Giải thuật) | 100% |
| Chương 4:  (100%) |  | **Xây dựng hệ thống:**   * Xây dựng phần mềm | 0% |
| Chương 5:  (100%) | 25/10 | **Kết luận và hướng phát triển:**   * Kết luận * TaskSheet(Bảng kế hoạch công việc) * Checklist(Bảng kiểm tra chức năng) | 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://chatgpt.com/>
      2. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

#### **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/>