**ĐẠI HỌC NGUYỄN TRÃI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙞🕮🙜---



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

***ĐỀ TÀI***

**TÊN ĐỀ TÀI**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:** | **Nguyễn Gia Duy** |
| **Mã sinh viên:** |  |
| **Lớp:** |  |
| **Khóa:** |  |

**@FITNTU.PJ** 12/2024

**ĐẠI HỌC NGUYỄN TRÃI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙞🕮🙜---



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

***ĐỀ TÀI***

**TÊN ĐỀ TÀI**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:** | **Nguyễn Gia Duy** |
| **Mã sinh viên:** | **2310900031** |
| **Lớp:** | **CNT3** |
| **Khóa:** | **K23** |

**@FITNTU.PJ** 12/2024

# **Lời cảm ơn**

Trước tiên, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến quý thầy **Th.S Trịnh Văn Chung** - người đã không ngừng tận tình giảng dạy và truyền đạt những kiến thức quý giá trong suốt quá trình học tập tại trường. Chính những kiến thức nền tảng và sự hướng dẫn tận tình từ thầy đã giúp em có được hành trang vững chắc để thực hiện báo cáo này.

Thầy cũng là người đã trực tiếp hỗ trợ, hướng dẫn và theo sát quá trình nghiên cứu của em. Những góp ý, chỉ dẫn tận tình và những kinh nghiệm quý báu mà thầy chia sẻ không chỉ giúp em hoàn thiện bài báo cáo, mà còn là nguồn cảm hứng lớn để em tiếp tục nỗ lực trong học tập và nghiên cứu. Sự kiên nhẫn và tâm huyết của thầy/cô là động lực rất lớn giúp em vượt qua những khó khăn trong suốt quá trình thực hiện.

Một lần nữa, em xin chân thành cảm ơn và kính chúc quý thầy cô thật nhiều sức khỏe, hạnh phúc và thành công trong công việc cũng như cuộc sống.

Contents

[**Lời cảm ơn** 1](#_Toc186814574)

[**Chương 1:** 3](#_Toc186814575)

[**NGHIÊN CỨU TÀ** 3](#_Toc186814576)

[**I LIỆU CƠ SỞ LÝ THUYẾT** 3](#_Toc186814577)

[**1.1.** **Tổng quan về lập trình thiết kế giao diện website:** 3](#_Toc186814578)

[**1.1.1. Giới thiệu về HTML:** 3](#_Toc186814579)

[**1.1.2. Giới thiệu về CSS3:** 5](#_Toc186814580)

[**1.1.3. Tổng quan về ngôn ngữ lập trình Javascript:** 5](#_Toc186814581)

[**1.1.4. Thư viện Jquery:** 7](#_Toc186814582)

[**1.1.5. Framework Bootstrap:** 9](#_Toc186814583)

[**1.2. Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server:** 9](#_Toc186814584)

[**1.2.1. Giới thiệu về SQL Server:** 9](#_Toc186814585)

[**1.2.2. Ưu điểm của SQL Server:** 12](#_Toc186814586)

[**1.2.3. Nhược điểm của SQL Server:** 13](#_Toc186814587)

[**1.3. Tổng quan về ngôn ngữ lập trình máy chủ:** 14](#_Toc186814588)

[**1.3.1. Giới thiệu về nền tảng công nghệ .NET:** 14](#_Toc186814589)

[**1.3.2. Giới thiệu về ngôn ngữ lập trình C#:** 17](#_Toc186814590)

[**1.3.3. Tổng quan về Framework ASP.NET MVC 5:** 20](#_Toc186814591)

[**Chương 2:** 24](#_Toc186814592)

[**PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG** 24](#_Toc186814593)

[**2.1. Người dùng của hệ thống:** 24](#_Toc186814594)

[**2.2. Chức năng của hệ thống:** 24](#_Toc186814595)

[**2.2.1. Systeme Designs (Thiết kế hệ thống):** 25](#_Toc186814596)

[**2.3. Database Design (Thiết kế cơ sở dữ liệu):** 26](#_Toc186814597)

# **Chương 1:**

# **NGHIÊN CỨU TÀ**

# **I LIỆU CƠ SỞ LÝ THUYẾT**

## **1.1.** [**Tổng quan về lập trình thiết kế giao diện website:**](https://docs.google.com/document/d/1nZ4QZdRXeQWBjhzZ1UiMlNBa6vbuchtg/edit#heading=h.2iq8gzs)

### **1.1.1. Giới thiệu về HTML:**

**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….

**Ưu điểm HTML:**

* Cộng đồng với nhiều tài nguyên lớn.
* Hoạt động mượt mà trên tất cả các trình duyệt.
* Các markup sử dụng trong HTML ngắn gọn và có độ đồng nhất cao.
* Sử dụng [mã nguồn mở](https://vietnix.vn/ma-nguon-mo-la-gi/) và miễn phí.
* Tích hợp dễ dàng với các ngôn ngữ backend (PHP, [Node.js](https://vietnix.vn/nodejs-la-gi/),..).

**Nhược điểm HTML:**

* Chủ yếu được áp dụng cho web tĩnh, nếu muốn tạo tính năng động thì các lập trình viên cần sử dụng thêm JavaScript hoặc [ngôn ngữ backend](https://vietnix.vn/backend-la-gi-frontend-la-gi/) khác.
* Mỗi trang HTML cần được tạo riêng biệt ngay khi có nhiều yếu tố trùng lặp như header và [footer](https://vietnix.vn/footer-la-gi/).
* Khó để kiểm soát cách đọc và hiển thị file HTML của trình duyệt (một số trình duyệt cũ không [render](https://vietnix.vn/render-la-gi/) được tag mới. Vì vậy, dù trong HTML document có sử dụng các tag thì các browser đó không đọc được).
* Một số trình duyệt còn chậm cập nhật để hỗ trợ tính năng mới của HTML**.**

**Các thẻ HTML cơ bản:**

* **Thẻ cấu trúc:**
  + <html>: Gói toàn bộ tài liệu HTML.
  + <head>: Chứa thông tin meta, tiêu đề, liên kết CSS, JavaScript.
  + <body>: Chứa nội dung chính hiển thị trên trình duyệt.
* **Thẻ tiêu đề:**
  + <title>: Đặt tiêu đề cho trang (hiển thị trên tab trình duyệt).
  + <h1> đến <h6>: Các mức tiêu đề, từ lớn nhất (<h1>) đến nhỏ nhất (<h6>).
* **Thẻ đoạn văn:**
  + <p>: Tạo đoạn văn bản.
* **Thẻ liên kết:**
  + <a href="url">: Tạo liên kết đến một địa chỉ URL hoặc một phần khác trong trang.
* **Thẻ hình ảnh:**
  + <img src="url" alt="Mô tả">: Hiển thị hình ảnh.
* **Thẻ danh sách:**
  + <ul>: Danh sách không có thứ tự (bullet points).
  + <ol>: Danh sách có thứ tự (đánh số).
  + <li>: Mục trong danh sách.
* **Thẻ bảng:**
  + <table>: Tạo bảng.
  + <tr>: Dòng (row) trong bảng.
  + <td>: Ô (cell) trong bảng.
  + <th>: Ô tiêu đề.
* **Thẻ định dạng văn bản:**
  + <b> hoặc <strong>: In đậm.
  + <i> hoặc <em>: In nghiêng.
  + <u>: Gạch dưới.

### **1.1.2. Giới thiệu về CSS3:**

**CSS3** là phiên bản thứ 3 và cũng là mới nhất của CSS, CSS3 được bổ sung thêm nhiều tính năng mới tiện lợi hơn CSS cho người dùng. Được thừa hưởng tất cả những gì có trong phiên bản trước và bổ sung các tính năng mới, CSS3 hiện rất được ưa chuộng trong thiết kế website.

**Ưu điểm CSS3:**

* **Tương thích với HTML5:** Khi mà HTML5 đang dần thay thế Flash, thì CSS3 chính là sự hỗ trợ cần thiết để có một giao diện website hoàn hảo.
* **Hiển thị cho các thiết bị có kích thước khác nhau:** Media Queries mới ra mắt trong CSS3 là bước ngoặt lớn cho các website. Hỗ trợ tương thích với các kích thước màn hình mà không cần chỉnh sửa nội dung hiển thị.
* **Tối ưu hóa công cụ tìm kiếm SEO:** Một điểm mạnh nữa của CSS3 được rất nhiều lập trình viên ưa chuộng là khả năng loại bỏ những đoạn code HTML bị thừa. Giúp các công cụ tìm kiếm có thể hoạt động tốt hơn.
* **Tương thích với mọi trình duyệt:** CSS3 cũng được đánh giá rất cao về khả năng tương thích khi có thể hoạt động tốt trên hầu hết các trình duyệt phổ biến. Dù hiển thị trên nhiều trình duyệt khác nhau nhưng website vẫn khá nhất quán.

### **1.1.3. Tổng quan về ngôn ngữ lập trình Javascript:**

**JavaScript** là một ngôn ngữ lập trình đa năng, chủ yếu được sử dụng để phát triển các tính năng động trên các trang web. Nó là một trong những công nghệ cốt lõi của lập trình web, cùng với HTML và CSS**.**

**JavaScript** là một trong những ngôn ngữ lập trình phổ biến và quan trọng nhất trong lĩnh vực phát triển web. Ban đầu, nó được thiết kế để thêm tính năng tương tác vào các trang web, nhưng giờ đây JavaScript đã phát triển thành một ngôn ngữ đa năng với khả năng hoạt động trên cả frontend và backend**.**

JavaScript (viết tắt là JS) là một ngôn ngữ lập trình kịch bản (scripting language), được sử dụng chủ yếu để tạo các tính năng tương tác trên các trang web.

Không giống như HTML và CSS, JavaScript cung cấp khả năng logic để xử lý dữ liệu, tương tác với người dùng, và kiểm soát hành vi của trang web.

**Đặc điểm nổi bật:**

* **Ngôn ngữ động (Dynamic):** Không cần khai báo kiểu dữ liệu trước khi sử dụng.
* **Chạy trên trình duyệt:** JavaScript được thực thi trực tiếp trong các trình duyệt web như Chrome, Firefox, Safari.
* **Đa năng:** Sử dụng được trên cả frontend (trình duyệt) và backend (Node.js).
* **Hỗ trợ lập trình hướng đối tượng (OOP):** Cho phép sử dụng các đối tượng và nguyên mẫu.

**Ưu điểm của ngôn ngữ lập trình Javascript:**

* Chương trình rất dễ học.
* Những lỗi Javascript rất dễ để phát hiện, từ đó giúp sửa lỗi một cách nhanh chóng hơn.
* Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.
* JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.
* Được các chuyên gia đánh giá là một loại ngôn ngữ lập trình nhẹ và nhanh hơn nhiều so với các ngôn ngữ lập trình khác.
* JS còn có thể được gắn trên một số các element hoặc những events của các trang web.
* Những website có sử dụng JS thì chúng sẽ giúp cho trang web đó có sự tương tác cũng như tăng thêm nhiều trải nghiệm mới cho người dùng.
* Người dùng cũng có thể tận dụng JS với mục đích là để kiểm tra những input thay vì cách kiểm tra thủ công thông qua hoạt động truy xuất database.
* Giao diện của ứng dụng phong phú với nhiều thành phần như Drag and Drop, Slider để cung cấp đến cho người dùng một Rich Interface (giao diện giàu tính năng).
* Giúp thao tác với người dùng phía Client và tách biệt giữa các Client với nhau.

**Nhược điểm của ngôn ngữ lập trình Javascript:**

* JS Code Snippet khá lớn.
* JS dễ bị các hacker và scammer khai thác hơn.
* JS cũng không có khả năng đa luồng hoặc đa dạng xử lý.
* Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.
* Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất.
* Vì tính bảo mật và an toàn nên các Client-Side **Javascript** sẽ không cho phép đọc hoặc ghi các file.
* JS không được hỗ trợ khi sử dụng ở trong tình trạng thiết bị được kết nối mạng.

### **1.1.4. Thư viện Jquery:**

**jQuery** là thư viện được viết từ JavaScript, jQuery giúp xây dựng các chức năng bằng Javascript dễ dàng, nhanh và giàu tính năng hơn được thiết kế để giúp đơn giản hóa việc viết mã JavaScript. Được phát hành lần đầu tiên vào năm 2006 bởi John Resig, jQuery cung cấp các API dễ sử dụng và hỗ trợ đa trình duyệt, giúp xử lý DOM, sự kiện, hiệu ứng và Ajax trở nên thuận tiện hơn.

**jQuery**được tích hợp nhiều module khác nhau. Từ module hiệu ứng cho đến module truy vấn selector. jQuery được sử dụng đến 99% trên tổng số website trên thế giới.

**Các module phổ biến của jQuery bao gồm:**

* Ajax – xử lý Ajax.
* Atributes – Xử lý các thuộc tính của đối tượng HTML.
* Effect – xử lý hiệu ứng.
* Event – xử lý sự kiện.
* Form – xử lý sự kiện liên quan tới form.
* DOM – xử lý Data Object Model.
* Selector – xử lý luồng lách giữa các đối tượng HTML.

**Các tính năng quan trọng của jQuery:**

* **Gọn nhẹ**: jQuery là một thư viện khá gọn nhẹ – có kích cỡ khoảng 19KB (gzipped).
* **Tương thích đa nền tảng**: Nó tự động sửa lỗi và chạy được trên mọi trình duyệt phổ biến như Chrome, Firefox, Safari, MS Edge, IE, Android và iOS.
* **Dễ dàng tạo Ajax**: Nhờ thư viện jQuery, code được viết bởi Ajax có thể dễ dàng tương tác với server và cập nhật nội dung tự động mà không cần phải tải lại trang.
* **Xử lý nhanh nhạy thao tác DOM**: jQuery giúp lựa chọn các phần tử DOM để traverse (duyệt) một cách dễ dàng, và chỉnh sửa nội dung của chúng bằng cách sử dụng Selector mã nguồn mở, mà còn được gọi là Sizzle.
* **Đơn giản hóa việc tạo hiệu ứng**: Giống với code snippet có hiệu ứng animation, nó phủ các dòng code và bạn chỉ việc thêm biến/nội dung vào thôi.
* **Hỗ trợ tốt phương thức sự kiện HTML**: Xử lý sự kiện − jQuery xử lý các sự kiện đa dạng mà không làm cho HTML code trở nên lộn xộn với các Event Handler.

**Ưu điểm của jQuery** thì vừa quan trọng vừa đáng giá:

* **jQuery** xử lý code rất nhanh và có khả năng mở rộng.
* **jQuery** tạo điều kiện cho người dùng viết các mã chức năng bằng các dòng tối thiểu.
* **jQuery** cải thiện hiệu suất lập trình web.
* **jQuery** phát triển các ứng dụng có tương thích với trình duyệt.

**Nhược điểm:**

* Hầu hết các tính năng mới của trình duyệt mới đều được **jQuery** sử dụng. Kích thước lớn hơn so với JavaScript thuần.
* Hiệu suất thấp hơn JavaScript thuần trong các dự án lớn.
* Đã bị thay thế một phần bởi các framework hiện đại như React, Angular, Vue

jQuery chứa đựng khá nhiều những thuật ngữ khác nhau, một số những thuật ngữ quan trọng nhất của jQuery cần biết:

* **jQuery Selectors:** được sử dụng để chọn và thao tác các phần tử HTML. Đây là phần rất quan trọng trong thư viện jQuery. Với jQuery Selectors, có thể tìm hoặc chọn các phần tử HTML dựa trên ID, classes, attributes, types và nhiều thứ khác từ DOM. Mỗi jQuery Selector bắt đầu bằng ký tự $ (). Kí tự này được gọi là factory function**.**
* **jQuery Tag Name:** Nó đại diện cho một tên thẻ có sẵn trong DOM.
* **Tag ID:** Tag ID đại diện cho một thẻ có sẵn với một ID cụ thể trong DOM.
* **Tag Class:** Tag Class đại diện cho một tag có sẵn với một class cụ thể trong DOM.
* **Function trong jQuery:**
* **Callback trong jQuery:** Một callback là một hàm thuần JavaScript được truyền một số phương thức như là một tham số hoặc tùy chọn. Một số callback là các sự kiện, được gọi để cung cấp cho người sử dụng cơ hội để phản ứng lại khi một trạng thái nào đó được kích hoạt. Hệ thống sự kiện trong jQuery sử dụng các callback này ở khắp mọi nơi.
* **Các Closure trong jQuery:** Các Closure được tạo bất cứ khi nào một biến được định nghĩa bên ngoài phạm vi hiện tại được truy cập từ bên trong phạm vi nội bộ.
* **Proxy Pattern trong jQuery:** Một Proxy có thể được sử dụng để điều khiển sự truy cập tới phần tử khác. Proxy thi hành cùng giao diện cho đối tượng khác này và truyền trên bất kỳ phương thức nào tới nó. Đối tượng khác này thường được gọi là Real Subject.Một Proxy có thể được khởi tạo tại vị trí của Real Subject này và cho phép nó để được truy cập ở chế độ từ xa. Chúng ta có thể lưu giữ phương thức setArray của jQuery trong một Closure và viết đè lên (overwrite) nó.
* **Phạm vi (Scope) trong jQuery:** Phạm vi của một biến là khu vực trong chương trình của bạn mà biến đó được định nghĩa. Biến trong JavaScript sẽ chỉ có hai phạm vi:
* **Các biến Global**: Một biến Global có phạm vi chung, nghĩa là nó được định nghĩa ở mọi nơi trong JavaScript code của bạn.
* **Các biến Local**: Một biến Local sẽ chỉ nhìn thấy bên trong một hàm nơi nó được định nghĩa. Các tham số hàm luôn luôn là Local cho hàm đó.
* **Các tham số trong jQuery:** Các tham số trong JavaScript là một loại của Array mà có thuộc tính length.
* **Context trong jQuery:** Từ khóa nổi tiếng trong JavaScript là this tham chiếu tới Context hiện tại. Trong một hàm, this có thể thay đổi, phụ thuộc cách hàm đó được gọi.

### **1.1.5. Framework Bootstrap:**

Bootstrap là một framework front-end mã nguồn mở, được thiết kế để giúp các nhà phát triển xây dựng giao diện web một cách nhanh chóng và hiệu quả. Được phát triển bởi Mark Otto và Jacob Thornton tại Twitter, Bootstrap được giới thiệu lần đầu vào năm 2011. Hiện nay, nó là một trong những framework phổ biến nhất cho thiết kế web

## **1.2. Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server:**

### **1.2.1. Giới thiệu về SQL Server:**

**SQL Server** chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

SQL Server là một hệ thống quản trị cơ sở dữ liệu (Database Management System - DBMS) sử dụng ngôn ngữ Structured Query Language (SQL) để tương tác với dữ liệu.

Nó hỗ trợ các thao tác như: lưu trữ, truy vấn, sửa đổi, xóa dữ liệu và xây dựng các cơ sở dữ liệu có tính phức tạp cao.

**Các thành cơ bản trong SQL Server gồm có**: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service,… Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

* **Database Engine**: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho ệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.
* **Integration Services**: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu.  Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,… và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.
* **Analysis Services**: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.
* **Notification Services**: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người dăng ký sử dụng trên nhiều loại thiết bị khác nhau.
* **Reporting  Services**: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.
* **Full Text Search Service**: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các [**cơ sở dữ liệu SQL.**](https://sqladvice.com/co-so-du-lieu-sql/)
* **Service Broker**: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

**Các phiên bản của SQL Server:** SQL Server có nhiều phiên bản phù hợp với nhu cầu của các tổ chức khác nhau:

* **Express Edition:** Miễn phí, dành cho các dự án nhỏ.
* **Standard Edition:** Dành cho doanh nghiệp vừa và nhỏ.
* **Enterprise Edition:** Đầy đủ tính năng, phù hợp cho các tổ chức lớn.
* **Developer Edition:** Dành cho lập trình viên phát triển và thử nghiệm.
* **Web Edition:** Tối ưu hóa cho các ứng dụng web.

**Kiến trúc của SQL Server:**

* **Cấu trúc cơ bản:** SQL Server có kiến trúc client-server, bao gồm:
* **Database Engine:** Thành phần chính chịu trách nhiệm lưu trữ, xử lý và bảo mật dữ liệu.
* **SQL Server Agent:** Tự động hóa các tác vụ quản trị, như sao lưu hoặc bảo trì.
* **Reporting Services (SSRS):** Hỗ trợ tạo báo cáo dữ liệu.
* **Integration Services (SSIS):** Tích hợp dữ liệu từ nhiều nguồn khác nhau.
* **Analysis Services (SSAS):** Phân tích dữ liệu và xây dựng hệ thống báo cáo.
* **Thành phần cơ bản trong cơ sở dữ liệu:**
* **Database:** Nơi lưu trữ dữ liệu.
* **Table:** Tổ chức dữ liệu thành các hàng (row) và cột (column).
* **Index:** Tăng tốc độ truy vấn dữ liệu.
* **View:** Một bảng ảo được tạo từ các truy vấn.
* **Stored Procedures:** Tập hợp các câu lệnh SQL được lưu trữ để tái sử dụng.
* **Trigger:** Tự động thực thi khi có một sự kiện xảy ra trong cơ sở dữ liệu.
* **Các tính năng nổi bật của SQL Server**
* **Khả năng quản lý dữ liệu lớn:** SQL Server hỗ trợ quản lý khối lượng dữ liệu lớn, bao gồm các tập dữ liệu có kích thước vài terabyte hoặc hơn.
* **Tính bảo mật cao:**
  + **Transparent Data Encryption (TDE):** Mã hóa toàn bộ cơ sở dữ liệu.
  + **Always Encrypted:** Mã hóa cột dữ liệu để bảo mật thông tin nhạy cảm.
  + **Row-Level Security:** Quản lý quyền truy cập ở mức hàng.
* **Khả năng mở rộng và hiệu suất cao:**
  + Hỗ trợ nhiều lõi CPU và bộ nhớ lớn.
  + **In-Memory OLTP:** Tăng hiệu suất xử lý dữ liệu giao dịch trực tuyến.
* **Hỗ trợ phân tích và báo cáo:**
  + **SSRS** giúp tạo báo cáo chi tiết và trực quan.
  + **SSAS** hỗ trợ phân tích dữ liệu theo nhiều chiều.
* **Tích hợp tốt với hệ sinh thái Microsoft:**
  + Hoạt động mượt mà với các sản phẩm khác như Power BI, Azure, và Microsoft Office.

### **1.2.2. Ưu điểm của SQL Server:**

* **Hiệu suất và khả năng mở rộng:**
* **Xử lý dữ liệu lớn:** SQL Server được thiết kế để xử lý khối lượng dữ liệu lớn với hiệu suất cao.
* **Tích hợp In-Memory OLTP:** Tăng tốc độ xử lý giao dịch lên đến hàng chục lần.
* **Hỗ trợ đa lõi CPU và bộ nhớ lớn:** Đảm bảo khả năng mở rộng khi nhu cầu xử lý tăng.
* **Bảo mật cao:**
* **Transparent Data Encryption (TDE):** Bảo mật dữ liệu bằng cách mã hóa toàn bộ cơ sở dữ liệu.
* **Always Encrypted:** Bảo vệ dữ liệu nhạy cảm ngay cả khi bị truy cập trái phép.
* **Row-Level Security:** Quản lý quyền truy cập dữ liệu chi tiết đến từng hàng.
* **Giao diện thân thiện:**
* **SQL Server Management Studio (SSMS):** Công cụ quản trị trực quan, dễ sử dụng, giúp quản lý cơ sở dữ liệu và thực thi truy vấn một cách thuận tiện.
* **Hỗ trợ tích hợp với Visual Studio:** Thuận lợi cho việc phát triển ứng dụng.
* **Hỗ trợ đa tính năng:**
* **Tích hợp BI:** Các công cụ như SSRS, SSAS, SSIS hỗ trợ phân tích và báo cáo dữ liệu mạnh mẽ.
* **Hỗ trợ các loại dữ liệu đa dạng:** Từ dữ liệu quan hệ (Relational Data) đến dữ liệu phi quan hệ (JSON, XML).
* **Khả năng quản lý giao dịch:** Hỗ trợ các tính năng như ACID để đảm bảo tính toàn vẹn dữ liệu.
* **Tích hợp tốt trong hệ sinh thái Microsoft:**
* **Azure Integration:** SQL Server dễ dàng tích hợp với Microsoft Azure, cung cấp các giải pháp điện toán đám mây mạnh mẽ.
* **Power BI và Excel:** Tương thích cao với các công cụ phân tích dữ liệu phổ biến của Microsoft.
* **Đa nền tảng:** Kể từ phiên bản SQL Server 2017, nó hỗ trợ hoạt động trên cả Windows và Linux, mở rộng sự lựa chọn cho các nhà phát triển.

### **1.2.3. Nhược điểm của SQL Server:**

* **Chi phí cao:**
* **Phiên bản trả phí đắt đỏ:** Các phiên bản như Standard hoặc Enterprise có chi phí bản quyền cao, đặc biệt đối với các doanh nghiệp vừa và nhỏ.
* **Chi phí duy trì:** Bao gồm phí hỗ trợ kỹ thuật và nâng cấp phiên bản.
* **Yêu cầu tài nguyên cao:**
* **Phần cứng:** SQL Server yêu cầu tài nguyên phần cứng mạnh mẽ để hoạt động hiệu quả, đặc biệt đối với các hệ thống lớn.
* **Tối ưu hóa:** Cần có kiến thức chuyên sâu để tối ưu hóa hiệu suất khi xử lý khối lượng dữ liệu lớn.
* **Phụ thuộc vào nền tảng Microsoft:**
* Mặc dù hỗ trợ Linux, SQL Server vẫn hoạt động tốt nhất trên hệ sinh thái Microsoft.
* Điều này có thể gây khó khăn nếu tổ chức sử dụng các công nghệ không phải của Microsoft.
* **Khả năng tích hợp với công nghệ khác hạn chế:** SQL Server không tích hợp tốt bằng các hệ quản trị cơ sở dữ liệu mã nguồn mở như MySQL hoặc PostgreSQL khi làm việc trong môi trường đa nền tảng hoặc với các công nghệ không phải của Microsoft.
* **Độ phức tạp trong quản trị:**
* **Đòi hỏi kiến thức chuyên môn:** Quản trị SQL Server có thể trở nên phức tạp khi hệ thống phát triển lớn, đòi hỏi kỹ năng từ các chuyên gia.
* **Khó khăn trong xử lý lỗi:** Một số lỗi phức tạp yêu cầu kiến thức sâu rộng để khắc phục.

**Tóm tắt:**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Hiệu suất cao, hỗ trợ dữ liệu lớn và khả năng mở rộng. | Chi phí bản quyền cao, đặc biệt cho các phiên bản đầy đủ tính năng. |
| Bảo mật mạnh mẽ với các tính năng như Always Encrypted và Row-Level Security. | Yêu cầu phần cứng mạnh và chi phí duy trì lớn. |
| Giao diện thân thiện, tích hợp tốt với các công cụ quản trị và phát triển của Microsoft. | Phụ thuộc nhiều vào hệ sinh thái Microsoft, tích hợp hạn chế với công nghệ không phải của Microsoft. |
| Hỗ trợ nhiều tính năng BI, phân tích và báo cáo. | Cần kỹ năng quản trị chuyên sâu để tối ưu hóa hệ thống lớn. |
| Hỗ trợ cả Windows và Linux. | Độ phức tạp trong việc khắc phục lỗi và xử lý sự cố. |

☞ SQL Server là một giải pháp mạnh mẽ và toàn diện cho quản lý cơ sở dữ liệu, phù hợp cho các tổ chức có yêu cầu cao về hiệu suất và bảo mật. Tuy nhiên, chi phí và yêu cầu phần cứng là những yếu tố cần cân nhắc trước khi triển khai.

## **1.3. Tổng quan về ngôn ngữ lập trình máy chủ:**

### **1.3.1. Giới thiệu về nền tảng công nghệ .NET:**

Chúng ta vẫn thường nhắc đến .NET như một ngôn ngữ lập trình. Nhưng trên thực tế, .NET không phải ngôn ngữ mà là một nền tảng lập trình nguồn mở ([open-source](https://vi.wikipedia.org/wiki/Ngu%E1%BB%93n_m%E1%BB%9F)) miễn phí và đa nền tảng. Nó được phát triển bởi Microsoft và được dùng để xây dựng các ứng dụng cho website, desktop, điện toán đám mây, game, app di động hay app AI.

.NET là một nền tảng thống nhất có khả năng tương tác với đa ngôn ngữ lập trình và được kiến trúc bởi 2 môi trường:

* Thư viện lớp lớn có tên là Framework Class Library (FCL). Môi trường cung cấp giao diện cho người dùng, kết nối cơ sở dữ liệu, mật mã, truy cập dữ liệu, truyền thông tin mạng, các thuật toán, thiết kế web và phát triển web.
* Thực thi các chương trình trong môi trường phần mềm Common Language Runtime (CLR). Một máy ảo cung cấp các dịch vụ như bảo mật, quản lý bộ nhớ và xử lý các ngoại lệ.

.NET framework được tạo ra nhằm mục đích hoàn thành mục tiêu:

* Cung cấp cho người dùng ngôn ngữ lập trình hướng đối tượng, tương thích hầu hết các mã lưu trữ và thực hiện của đối tượng.
* Giảm thiểu triển khai phần mềm và xung đột, thúc đẩy việc thực hiện mã an toàn. Nó đồng thời giúp loại bỏ các vấn đề hiệu suất của các môi trường kịch bản hoặc diễn giải bằng cách cung cấp môi trường thực thi mã.
* Tạo nên trải nghiệm nhất quán cho các nhà phát triển trên nhiều ứng dụng khác nhau.
* Khả năng tích hợp bất kỳ mã khác giúp xây dựng được tất cả thông tin liên lạc và tiêu chuẩn ngành công nghiệp.

NET đã được phát triển với rất nhiều phiên bản. Mới nhất là .NET 5.0 – được kế thừa từ phiên bản .NET Core 3.1 và .NET Framework 4.6.

**.NET Framework:** .NET Framework là nền tảng phát triển phần mềm ban đầu của Microsoft, được thiết kế để hỗ trợ việc phát triển ứng dụng Windows. Nó bao gồm một thư viện lớp (class library) phong phú và môi trường thực thi (CLR - Common Language Runtime). .NET Framework chủ yếu được sử dụng trên hệ điều hành Windows.

**.NET Core:** .NET Core là phiên bản hiện đại và mã nguồn mở của .NET Framework, được Microsoft phát triển để hỗ trợ đa nền tảng (Windows, Linux, macOS).

.NET Core cung cấp khả năng phát triển các ứng dụng web, dịch vụ API và các ứng dụng đám mây.

**.NET 5 và các phiên bản tiếp theo:** .NET 5 là sự kế thừa của .NET Core, đánh dấu sự chuyển đổi mạnh mẽ với sự kết hợp các công nghệ từ .NET Framework và .NET Core. Các phiên bản sau như .NET 6, .NET 7, và dự kiến .NET 8 tiếp tục cải tiến và mở rộng khả năng của nền tảng này.

**Xamarin:** Xamarin là một công nghệ phát triển ứng dụng di động đa nền tảng, cho phép các lập trình viên viết ứng dụng di động cho cả **iOS** và **Android** bằng C# và .NET.

**Blazor:** Blazor là một framework mới của .NET cho phép viết ứng dụng web interative trên phía client bằng C# thay vì JavaScript.

**Ưu Điểm:**

* **Đa nền tảng:** .NET Core và .NET 5 trở đi hỗ trợ phát triển ứng dụng cho cả Windows, macOS và Linux, giúp mở rộng khả năng sử dụng.
* **Cộng đồng và hỗ trợ mạnh mẽ:** .NET có cộng đồng lớn và tài liệu phong phú từ Microsoft, giúp lập trình viên dễ dàng tìm kiếm hỗ trợ.
* **Hiệu suất cao:** .NET Core và .NET 5+ tối ưu hóa hiệu suất với các tính năng như xử lý đa luồng, giảm thiểu chi phí bộ nhớ và giảm độ trễ.
* **Tính tương thích ngược (Backward Compatibility):** Các phiên bản mới của .NET vẫn duy trì khả năng tương thích với các ứng dụng và thư viện đã xây dựng trên các phiên bản cũ hơn của .NET Framework.
* **Bảo mật mạnh mẽ:** .NET cung cấp các tính năng bảo mật như xác thực, mã hóa và bảo vệ dữ liệu tích hợp sẵn.

**Nhược Điểm:**

* **Chi phí bản quyền với Windows:** Mặc dù .NET Core hỗ trợ Linux và macOS, nhưng việc phát triển các ứng dụng trên nền tảng Windows vẫn yêu cầu bản quyền cho hệ điều hành.
* **Khó khăn khi phát triển ứng dụng với GUI (giao diện người dùng) trên Linux/macOS:** Các ứng dụng GUI phát triển bằng Windows Forms hoặc WPF có thể gặp khó khăn trong việc chạy trên các nền tảng không phải Windows.
* **Tính khả dụng của thư viện bên thứ ba:** Mặc dù .NET có thư viện phong phú, nhưng một số thư viện bên thứ ba phổ biến lại chưa được hỗ trợ tốt trên .NET Core.

**Các Ngôn Ngữ Lập Trình trong .NET:**

* **C#:** C# là ngôn ngữ lập trình chính và phổ biến nhất trong .NET, được thiết kế với cú pháp dễ hiểu và mạnh mẽ, phù hợp với cả phát triển ứng dụng desktop, web và di động.
* **VB.NET:** VB.NET là một ngôn ngữ lập trình khác trong nền tảng .NET, có cú pháp gần gũi với ngôn ngữ Visual Basic. Tuy nhiên, nó có tính năng mạnh mẽ như C# trong việc phát triển ứng dụng.
* **F#:** F# là một ngôn ngữ lập trình hàm được phát triển trong .NET, hỗ trợ việc xử lý bất đồng bộ, lập trình hàm, và toán học phức tạp.

### **1.3.2. Giới thiệu về ngôn ngữ lập trình C#:**

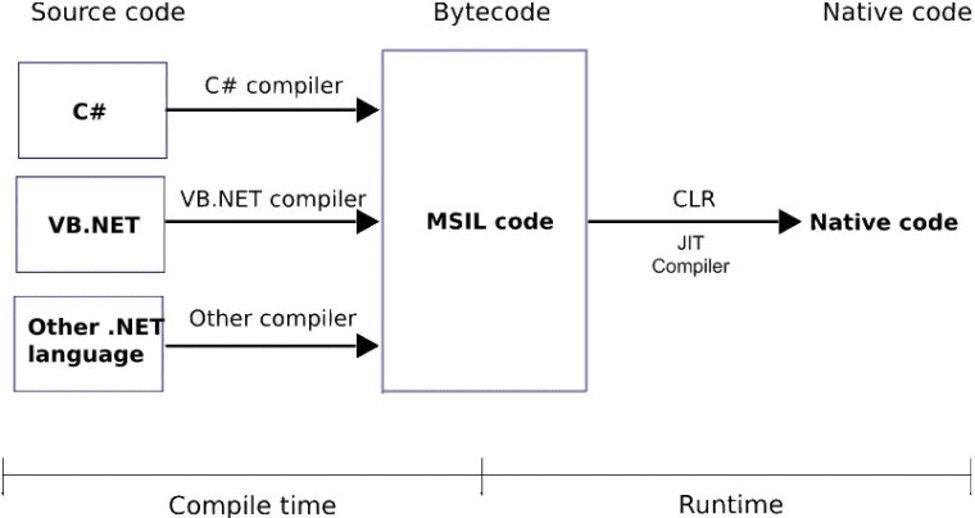
**C#** (đọc là C-sharp) là một ngôn ngữ lập trình đa năng, hướng đối tượng, được phát triển bởi **Microsoft** trong khuôn khổ nền tảng .NET. Được thiết kế để đơn giản, hiện đại và mạnh mẽ, C# chủ yếu được sử dụng để phát triển các ứng dụng desktop, web, di động, và trò chơi. C# cũng là một trong những ngôn ngữ chủ đạo trong hệ sinh thái .NET, giúp lập trình viên tạo ra phần mềm hiệu quả và dễ duy trì.

**C#** được giới thiệu lần đầu tiên vào năm 2000, dưới sự dẫn dắt của Anders Hejlsberg (người sáng lập Delphi). C# mang những đặc điểm tốt nhất từ các ngôn ngữ lập trình khác như C++ và Java, kết hợp với cú pháp đơn giản và các tính năng mạnh mẽ, giúp các lập trình viên xây dựng phần mềm dễ dàng và nhanh chóng.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.



C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

**Đặc điểm chính của C#:**

* **Hướng đối tượng (OOP)**: C# hỗ trợ các nguyên lý hướng đối tượng như kế thừa, đa hình, đóng gói và trừu tượng hóa.
* **An toàn với kiểu dữ liệu**: C# sử dụng hệ thống kiểu dữ liệu mạnh mẽ, giúp phát hiện lỗi khi biên dịch thay vì khi chạy.
* **Quản lý bộ nhớ tự động**: C# sử dụng garbage collection để tự động quản lý bộ nhớ, giúp giảm bớt gánh nặng cho lập trình viên khi quản lý tài nguyên bộ nhớ.
* **Tính tương thích đa nền tảng**: Kể từ khi .NET Core ra đời, C# đã có thể chạy trên nhiều nền tảng như Windows, Linux và macOS.
* **Mã nguồn mở**: C# và .NET Core đều là mã nguồn mở, có thể được phát triển và cải tiến bởi cộng đồng lập trình viên toàn cầu.

**Các Tính Năng Nổi Bật của C#:**

* **Hướng đối tượng:**
* **Lớp (Classes)** và **đối tượng (Objects)**: C# hỗ trợ định nghĩa các lớp và đối tượng để tổ chức mã nguồn theo kiểu hướng đối tượng.
* **Kế thừa (Inheritance)**: C# cho phép một lớp con kế thừa thuộc tính và phương thức từ lớp cha, tái sử dụng mã nguồn và mở rộng chức năng.
* **Đa hình (Polymorphism)**: Các phương thức có thể được định nghĩa lại trong các lớp con để thay đổi hành vi của chúng.
* **Đóng gói (Encapsulation)**: Các thuộc tính và phương thức của lớp có thể được bảo vệ bằng cách sử dụng các mức độ truy cập (public, private, protected).
* **Các tính năng mạnh mẽ:**
* **Các delegate và sự kiện**: C# hỗ trợ các delegate (tham chiếu đến phương thức) và sự kiện, giúp dễ dàng xây dựng các ứng dụng phản ứng với các thay đổi trong trạng thái hoặc sự kiện.
* **LINQ (Language Integrated Query)**: C# hỗ trợ LINQ, giúp truy vấn dữ liệu một cách dễ dàng ngay trong ngữ cảnh ngôn ngữ lập trình, giúp truy vấn mảng, danh sách, cơ sở dữ liệu và XML.
* **Tính đồng bộ và bất đồng bộ (async/await)**: C# cung cấp cú pháp đơn giản cho lập trình bất đồng bộ, giúp tối ưu hóa hiệu suất của các ứng dụng có nhiều tác vụ I/O.
* **Các thuộc tính (Attributes)**: C# cho phép gắn các thông tin thêm vào lớp, phương thức hoặc trường với các thuộc tính, giúp hỗ trợ trong việc tự động hóa mã và lập trình phản xạ.
* **Quản lý bộ nhớ:**
* **Garbage Collection**: C# tự động quản lý bộ nhớ thông qua cơ chế garbage collection, giúp giải phóng bộ nhớ mà không cần lập trình viên phải trực tiếp quản lý.
* **Không cần chỉ định con trỏ**: C# không yêu cầu người lập trình phải thao tác với con trỏ (pointer), giúp giảm thiểu lỗi và nguy cơ tràn bộ nhớ.
* **Môi trường phát triển**:
* **Visual Studio**: Là công cụ phát triển chính cho C#, cung cấp nhiều tính năng như gỡ lỗi, kiểm tra mã, IntelliSense, và công cụ hỗ trợ kiểm thử.
* **.NET Framework và .NET Core**: C# làm việc rất tốt với .NET, giúp phát triển ứng dụng trên nhiều nền tảng, từ ứng dụng Windows cho đến các ứng dụng di động và web.

**Các Ứng Dụng Phổ Biến của C#**: C# là một ngôn ngữ đa năng, được sử dụng trong nhiều lĩnh vực khác nhau:

* **Phát triển ứng dụng Desktop:** C# thường được sử dụng để phát triển các ứng dụng Windows với **Windows Forms** và **WPF (Windows Presentation Foundation)**.
* **Phát triển ứng dụng Web: ASP.NET** là một framework mạnh mẽ cho phép lập trình viên phát triển các ứng dụng web động, dịch vụ API, và các ứng dụng thương mại điện tử.
* **Phát triển ứng dụng di động: Xamarin** cho phép phát triển ứng dụng di động chạy trên cả **iOS** và **Android** với mã nguồn chung, giúp giảm thiểu chi phí và thời gian phát triển.
* **Phát triển game: Unity** là một engine game sử dụng C# làm ngôn ngữ chính để phát triển các trò chơi 2D và 3D. Unity hỗ trợ nhiều nền tảng như PC, console, di động và thực tế ảo (VR).
* **Ứng dụng đám mây và dịch vụ API:** C# và **ASP.NET Core** thường được sử dụng để xây dựng các ứng dụng đám mây và dịch vụ API, đặc biệt trong môi trường **Azure** của Microsoft.

**Ưu Điểm:**

* **Đơn giản và dễ học**: C# có cú pháp đơn giản và dễ đọc, dễ học cho cả người mới bắt đầu và lập trình viên có kinh nghiệm.
* **Đa nền tảng**: C# hỗ trợ phát triển ứng dụng chạy trên nhiều nền tảng, từ Windows, Linux, macOS cho đến di động.
* **Tính năng mạnh mẽ**: C# cung cấp các tính năng như LINQ, async/await, và delegate giúp việc phát triển phần mềm trở nên nhanh chóng và dễ dàng.
* **Hỗ trợ mạnh mẽ từ Microsoft**: Là ngôn ngữ của Microsoft, C# có sự hỗ trợ tuyệt vời từ các công cụ phát triển như **Visual Studio** và các tài liệu học tập đầy đủ.
* **Cộng đồng phát triển lớn**: C# có cộng đồng phát triển rộng lớn, với rất nhiều tài nguyên và hỗ trợ từ các lập trình viên toàn cầu.

**Nhược điểm:**

* **Quá phụ thuộc vào Microsoft**: Mặc dù C# và .NET Core hỗ trợ đa nền tảng, nhưng một số công nghệ của C# vẫn phụ thuộc vào hệ sinh thái của Microsoft.
* **Chạy trên nền tảng di động chậm hơn một chút**: Mặc dù Xamarin giúp phát triển ứng dụng di động bằng C#, nhưng hiệu suất của các ứng dụng di động trên C# có thể chậm hơn so với các công nghệ như Swift (iOS) hoặc Kotlin (Android).
* **Không phổ biến như Java hoặc JavaScript**: Mặc dù C# rất mạnh mẽ, nhưng nó không phổ biến bằng các ngôn ngữ như JavaScript, Java hoặc Python trong một số lĩnh vực.

### **1.3.3. Tổng quan về Framework ASP.NET MVC 5:**

**ASP.NET MVC 5** là một framework web mạnh mẽ và linh hoạt được phát triển bởi Microsoft, thuộc phần của nền tảng ASP.NET, giúp xây dựng các ứng dụng web dựa trên mô hình Model-View-Controller (MVC). MVC 5 được thiết kế để phát triển các ứng dụng web hiện đại, hỗ trợ tính năng mạnh mẽ, dễ duy trì và dễ kiểm soát trong quá trình phát triển.

**ASP.NET MVC 5** là phiên bản nâng cấp của ASP.NET MVC 4, với nhiều cải tiến và tính năng mới, bao gồm hỗ trợ tính năng mới trong việc xử lý routing, authentication và authorization, cũng như sự tích hợp với các công nghệ mới.

**Các Tính Năng Chính Của ASP.NET MVC 5**

* **Routing tinh vi:**
* Routing trong ASP.NET MVC 5 giúp xác định và quản lý các URL của ứng dụng. Nó xác định cách các yêu cầu HTTP được chuyển hướng tới các controller và action cụ thể.
* ASP.NET MVC 5 hỗ trợ tính năng Attribute Routing, cho phép lập trình viên chỉ định các route trực tiếp trong các phương thức controller bằng cách sử dụng các thuộc tính (attributes). Điều này giúp dễ dàng quản lý các URL tùy chỉnh mà không cần phải thay đổi cấu hình routing trong tệp RouteConfig.cs.
* **Hỗ trợ Mobile (Responsive Design):**
* ASP.NET MVC 5 tích hợp tính năng Mobile-First Design, giúp các ứng dụng web hiển thị tốt trên tất cả các thiết bị từ desktop đến mobile.
* Framework này hỗ trợ Media Queries để phát triển các giao diện người dùng thân thiện với thiết bị di động.
* **Hỗ trợ Authentication và Authorization:** ASP.NET MVC 5 hỗ trợ các phương pháp xác thực và phân quyền (authentication and authorization) mạnh mẽ, bao gồm:
  + Identity Framework: Quản lý người dùng, quyền truy cập và xác thực thông qua tài khoản người dùng, hỗ trợ đăng nhập bằng các phương thức như Google, Facebook, Twitter, và Microsoft.
  + OAuth và OpenID Connect: Các giao thức phổ biến cho xác thực người dùng qua bên thứ ba, giúp bảo mật thông tin người dùng và cải thiện khả năng sử dụng.
* **Hỗ trợ các tính năng mới trong AJAX:**
* ASP.NET MVC 5 tích hợp với jQuery để hỗ trợ các tính năng AJAX (Asynchronous JavaScript and XML), giúp tải dữ liệu từ máy chủ và cập nhật giao diện người dùng mà không cần tải lại trang.
* AJAX Helpers giúp các nhà phát triển dễ dàng gọi các phương thức controller từ view mà không phải viết mã JavaScript phức tạp.
* **Kết nối tốt với Entity Framework:**
* ASP.NET MVC 5 hỗ trợ Entity Framework (EF), một công cụ ORM (Object-Relational Mapping) giúp dễ dàng làm việc với cơ sở dữ liệu.
* EF giúp chuyển đổi dữ liệu giữa các đối tượng trong C# và cơ sở dữ liệu mà không cần phải viết nhiều mã SQL thủ công.
* **Hỗ trợ Bundling và Minification:**
* Bundling và Minification là các tính năng giúp tối ưu hóa tài nguyên của ứng dụng, như các tập tin CSS và JavaScript, để giảm thiểu dung lượng tải về và cải thiện hiệu suất của ứng dụng web.
* ASP.NET MVC 5 cung cấp các công cụ để gom nhóm (bundle) các tệp tài nguyên và rút gọn (minify) chúng trong quá trình biên dịch, giúp giảm thời gian tải trang và tiết kiệm băng thông.
* **Hỗ trợ Dependency Injection:**
* Dependency Injection (DI) là một kỹ thuật thiết kế giúp giảm sự phụ thuộc giữa các lớp và dễ dàng kiểm thử, bảo trì mã.
* ASP.NET MVC 5 hỗ trợ DI thông qua các thư viện bên ngoài như Ninject, Unity, Autofac, hoặc sử dụng DI sẵn có trong ASP.NET Core.

**Quản lý cấu hình và các công cụ hỗ trợ:**

* **Công cụ Visual Studio:** Visual Studio là công cụ phát triển chính thức được khuyến nghị để phát triển ứng dụng ASP.NET MVC 5, cung cấp một môi trường phát triển tích hợp (IDE) mạnh mẽ với tính năng IntelliSense, gỡ lỗi và kiểm tra mã.
* **Scaffolding:** Scaffolding là tính năng tự động tạo mã cho các ứng dụng web, giúp tạo ra các controller, view và mô hình (model) một cách nhanh chóng, tiết kiệm thời gian trong quá trình phát triển.

**Ưu Điểm Của ASP.NET MVC 5:**

* **Kiến trúc rõ ràng:** Với mô hình MVC, ASP.NET MVC 5 giúp tách biệt rõ ràng giữa các thành phần của ứng dụng (model, view, controller), giúp mã nguồn dễ duy trì và mở rộng.
* **Dễ dàng mở rộng và tùy chỉnh:**Framework cung cấp các điểm mở rộng và khả năng tùy chỉnh cao, cho phép lập trình viên thay đổi và mở rộng các chức năng của ứng dụng mà không ảnh hưởng đến các phần còn lại.
* **Tích hợp mạnh mẽ với các công nghệ khác:** ASP.NET MVC 5 dễ dàng tích hợp với các công nghệ khác như Entity Framework, Web API, SignalR và jQuery để phát triển các ứng dụng web động và mạnh mẽ.
* **Quản lý bảo mật tốt:** Với các tính năng bảo mật tiên tiến, như bảo mật mã hóa, quản lý người dùng và phân quyền (authentication and authorization), ứng dụng web trong ASP.NET MVC 5 có thể bảo vệ dữ liệu người dùng một cách hiệu quả.

**Nhược Điểm của ASP.NET MVC 5:**

* Khó học đối với người mới bắt đầu: ASP.NET MVC yêu cầu một số kiến thức cơ bản về mô hình MVC và C#. Với những người chưa có kinh nghiệm về phát triển web hoặc lập trình hướng đối tượng, việc làm quen với ASP.NET MVC có thể khá khó khăn.
* Tài liệu hướng dẫn chưa đầy đủ: Dù có rất nhiều tài liệu hỗ trợ, nhưng đôi khi các tài liệu hướng dẫn có thể chưa đầy đủ hoặc không cập nhật nhanh chóng với các tính năng mới trong phiên bản ASP.NET MVC.
* Không phải là lựa chọn tốt cho các ứng dụng đơn giản: ASP.NET MVC 5 có thể quá phức tạp và dư thừa đối với các ứng dụng web đơn giản, nơi mà các giải pháp khác như ASP.NET Web Forms hoặc Blazor có thể phù hợp hơn.

# **Chương 2:**

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

## **2.1. Người dùng của hệ thống:**

Đối tượng người dùng của hệ thống bao gồm: Admin (Quản trị hệ thống).

## **2.2. Chức năng của hệ thống:**

Admin functions (Chức năng dành cho quản trị viên): có những chức năng 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 như: tên khách hàng, địa chỉ, số điện thoại,…  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 thể quản lí các sản phẩm trong các danh mục có trong hệ thống |
| Nhập | Click vào danh mục muốn thực hiện thao tác |
| Xử lý | Thực hiện các thao tác thành công |
| Xuất | Chuyển đến trang quản trị muốn tới |

* Chức năng thêm mới sửa xóa các sản phẩm:

|  |  |
| --- | --- |
| Mô tả | Quản trị viên click đến bảng cần sửa , xem thông tin |
| Nhập | Chỉnh sửa theo ý muốn của quản trị viên |
| Xử lý | Thực hiện các thao tác thành công |
| Xuất | Cập nhật thành công các thao tác đã làm |

### **2.2.1. Systeme Designs (Thiết kế hệ thống):**

Thực thể quản trị (**ngd\_quantri**): Lưu trữ thông tin người quản trị viên hệ thống.

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

Thực thể khách hàng (**ngd\_sanpham**): Hiển thị các sản phẩm có trong hệ thống.

Thực thể khách hàng (**ngd\_loaisanpham**): Phân chia các loại sản phẩm phù hợp.

Thực thể khách hàng (**ngd\_hoadon**): Hiển thị các hóa đơn của khách hàng.

Thực thể khách hàng (**ngd\_cthoadon**): Hiển thị chi tiết các hóa đơn của khách hàng.

## **2.3. Database Design (Thiết kế cơ sở dữ liệu):**

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

* Bảng ngd\_quantri:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột  Field | Kiểu dữ liệu DataType | Null | Ràng buộc  Constraint | Mô tả Content |
| ngdTaiKhoan | Varchar(50) | No | Primary Key | Tài khoản đăng nhập của quản trị |
| ngdMatKhau | Varchar(32) | No |  | Mật khẩu đăng nhập của quản trị. |
| ngdTrangThai | Tinyint |  |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

* Bảng ngd\_khachhang:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột  Field | Kiểu dữ liệu DataType | Null | Ràng buộc  Constraint | Mô tả Content |
| Id | INT | No | Primary Key; Auto\_increment | Id khach hàng | |
| ngdMaKhachHang | Int | No |  | Mã khách hàng | |
| ngdHoTenKhachHang | Varchar(255) | Yes |  | Tên khách hàng | |
| ngdEmail | Varchar(255) | No |  | Hộp thư điện tử | |
| ngdMatKhau | Varchar(255) | Yes |  | Mật khẩu | |
| ngdDienThoai | Varchar(255) | Yes |  | Điện thoại | |
| ngdDiaChi | Varchar(255) | Yes |  | Địa chỉ | |
| ngdNgayDangKi | DateTime | Yes | Current | Ngày cập nhật vào hệ thống | |
| ngdTrangThai | Tinyint | Yes |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa | |

* Bảng ngd\_sanpham:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột  Field | Kiểu dữ liệu DataType | Null | Ràng buộc  Constraint | Mô tả Content |
| Id | INT | No | Primary Key | Tài khoản đăng nhập của quản trị |
| ngdMaSanPham | Varchar(255) | No |  | Mật khẩu đăng nhập của quản trị. |
| ngdTenSanPham | Varchar(255) | No |  | Tên sản phâm |
| ngdHinhAnh | Varchar(255) | No |  | Hình ảnh sản phẩm |
| ngdSoLuong | INT | No |  | Số lượng sản phẩm |
| ngdDonGia | Double | No |  | Giá sản phẩm |
| ngdMaLoai | INT | No |  | Ma loại sản phẩm |
| ngdTrangThai | Tinyint |  |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

* Bảng ngd\_loaisanpham:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột  Field | Kiểu dữ liệu DataType | Null | Ràng buộc  Constraint | Mô tả Content |
| Id | INT | No | Primary Key | Id loại sản phẩm |
| ngdMaLoai | Varchar(255) | No |  | Mã loại sản phẩm |
| ngdTenLoai | Varchar(255) | No |  | Tên loại sản phẩm |
| ngdTrangThai | Varchar(255) | No |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

* Bảng ngd\_hoadon:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột  Field | Kiểu dữ liệu DataType | Null | Ràng buộc  Constraint | Mô tả Content |
| Id | INT | No | Primary Key | Id hóa đơn |
| ngdMaHoaDon | Varchar(255) | No |  | Mã hóa đơn |
| ngdKhachHang | Varchar(255) | No |  | Thông tin khách hàng |
| ngdNgayHoaDon | Date | No |  | Ngày nhận hóa đơn |
| ngdHoTenKhachHang | Varchar(255) | No |  | Họ tên khách hàng |
| ngdEmail | Varchar(255) | No |  | Địa chỉ email khách hàng |
| ngdDienThoai | Varchar(255) | No |  | Số điện thoại khách hàng |
| ngdDiaChi | Varchar(255) | No |  | Địa chỉ khách hàng |
| ngd\_TongTriGia | Double | No |  | Trị giá sản phẩm |
| ngdTrangThai | Tinyint | No |  | Trạng thái:  1- Đang hoạt động 0- Đang bị khóa |

* Bảng ngd\_cthoadon:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột  Field | Kiểu dữ liệu DataType | Null | Ràng buộc  Constraint | Mô tả Content |
| ngdHoaDonID | INT | No | Primary Key | ID hóa đơn |
| ngdSanPhamID | INT | No |  | ID sản phẩm |
| ngdSoLuongMua | INT | No |  | Số lượng đã bán |
| ngdDonGia | Double | No |  | Đơn giá |
| ngdThanhTien | Double | No |  | Số tiền cần thanh toán |
| ngdTrangThai | Tinyint | No |  | Hoàn thành  Chưa hoàn thành |

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* **Kết luận**

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

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

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

* ….
* ….
* **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)**  
Kế Hoạch Project1

Họ Và Tên: Nguyễn Gia Duy      Mã Sinh Viên:2310900031.

Lớp K23CNT3                                 Khóa Học: K23

Đề Tài: Quản lí trang web

Kế Hoạch Thực Hiện:

|  |  |  |
| --- | --- | --- |
| Thời Gian | Nội Dung Thực Tập | Ghi Chú |
| Buổi 1:20/11/2024 | Hoàn Thành Nội Dung Chương 1 và 2 | Hoàn Thành Chương 1 và 2 |
| Buổi 2:25/11/2024 | Hoàn thành lab02 | https://github.com/Nguyen-Gia-Duy/Project1\_lab2 |
| Buổi 3:27/11/2024 | Hoàn Thành lab04 blade | https://github.com/Nguyen-Gia-Duy/nguyengiaduy-ls04 |
| Buổi 5:6/12/2024 | Hoàn Thành lab05 validation và request | https://github.com/Nguyen-Gia-Duy/project-lab05 |
| Buổi 6:9/12/2024 | Hoàn Thành Lab06 Session |  |
| Buổi 7:10/12/2024 | Lab07 Database |  |
| Buổi 8:11/12/2024 | Hoàn thành Lab08 QueryBuider | https://github.com/Nguyen-Gia-Duy/Nguyen-gia-duy-lab08 |
| Buổi 9:13/12/2024 | Hoàn Thành Lab09 Model | https://github.com/Nguyen-Gia-Duy/nguyen-gia-duy-ls09 |
| Buổi 10:16/12/2024 | Hoàn Thành Lab10  Migrations | https://github.com/Nguyen-Gia-Duy/nguyen-gia-duy-ls10 |
| Buổi 11:17/12/2024 | Hoàn Thành Bài Tập: Thực hành các chức năng: CRUD trên các bảng đã tạo   tvcNhaCC  tvcVattu | https://github.com/Nguyen-Gia-Duy/NguyenGiaDuy-ls11 |
| Buổi 12:18/12/2024 | Migrate database theo csdl cung cấp Đặt tên theo quy tắc | https://github.com/Nguyen-Gia-Duy/NGuyenGiaDUy-lesson12 |
| Buổi 14:23/12/2024 (chiều) | Tạo NGD\_SAN\_PHAM và hoàn thành CRUD | https://github.com/Nguyen-Gia-Duy/nguyen-gia-duy-ls14-15 |
| 26/12/2024 | Hoàn thành Admins | https://github.com/Nguyen-Gia-Duy/Nguyen-Gia-Duy-ls15 |

**TÀI LIỆU THAM KHẢO:**

**Tài liệu:**

**Website:**

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