

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ 1 NĂM HỌC 2024 – 2025

**TÌM HIỂU NGÔN NGỮ XML VÀ THIẾT KẾ
CẤU TRÚC LƯU TRỮ DỮ LIỆU SINH VIÊN
MINH HỌA**

Giáo viên hướng dẫn:

ThS.Nguyễn Ngọc Đan Thanh

Sinh viên thực hiện:

Họ tên: Tô Văn Tới

MSSV: 110121252

Lớp: DA21TTA

Trà Vinh, tháng 01 năm 2024

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ 1 NĂM HỌC 2024 – 2025

**TÌM HIỂU NGÔN NGỮ XML VÀ THIẾT KẾ
CẤU TRÚC LƯU TRỮ DỮ LIỆU SINH VIÊN
MINH HỌA**

Giáo viên hướng dẫn:

ThS.Nguyễn Ngọc Đan Thanh

Sinh viên thực hiện:

Họ tên: Tô Văn Tới

MSSV: 110121252

Lớp: DA21TTA

Trà Vinh, tháng 01 năm 2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

1. Quá trình thực hiện:

- Nghiêm túc, báo cáo tiến độ theo lịch
- Có khả năng nghiên cứu độc lập

2. Về báo cáo

- Đúng mẫu
- Đầy đủ các nội dung theo đề cương

3. Về chương trình

- Chương trình cơ bản đáp ứng theo yêu cầu đã phân tích
- Dữ liệu minh họa phong phú

4. Kết luận: Đạt mức Khá

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày tháng năm

Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

[illegible]

Trà Vinh, ngày tháng năm

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Gửi Cô Nguyễn Ngọc Đan Thanh,

Trong suốt quá trình thực hiện đồ án này, tôi đã nhận được sự hướng dẫn tận tình, chi tiết và đầy đủ từ Cô. Lời đầu tiên, tôi xin bày tỏ lòng biết ơn sâu sắc nhất đến Cô vì đã luôn kiên nhẫn, nhiệt tình trong việc truyền đạt kiến thức và kinh nghiệm quý báu, giúp tôi vượt qua những khó khăn và thách thức trong quá trình học tập và nghiên cứu.

Những lời khuyên, góp ý của Cô không chỉ là kim chỉ nam cho sự phát triển của đồ án mà còn là nguồn động viên, khích lệ tinh thần lớn lao cho tôi. Cô không chỉ là người thầy, mà còn là người bạn, người đồng hành cùng tôi trong suốt hành trình làm đồ án này.

Tôi xin cam kết sẽ tiếp tục nỗ lực không ngừng để không phụ lòng Cô đã dành cho tôi. Những kiến thức và kinh nghiệm mà Cô đã truyền đạt không chỉ giúp tôi hoàn thành tốt đồ án mà còn là hành trang vô giá cho con đường sự nghiệp tương lai của tôi.

Tôi rất may mắn và tự hào khi được học hỏi dưới sự hướng dẫn của Cô. Một lần nữa, tôi xin chân thành cảm ơn Cô vì tất cả những điều tốt lành mà Cô đã mang đến cho quá trình học tập và sự phát triển của tôi.

Xin chân thành cảm ơn Cô.

Trà Vinh, tháng 01 năm 2024

Tô Văn Tới

MỤC LỤC

TÓM TẮT ĐỒ ÁN THỰC TẬP CƠ SỞ NGÀNH	7
MỞ ĐẦU	8
CHƯƠNG 1 TỔNG QUAN	10
1.1 Tổng quan về vấn đề	10
1.2 Cách giải quyết	10
CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT	12
2.1 Ngôn ngữ XML	12
2.1.1 Giới thiệu tổng quan	12
2.1.1.1 Phân tử gốc và phân tử con	12
2.1.1.2 Thuộc tính	12
2.1.1.3 Ví dụ cấu trúc lưu trữ dữ liệu	13
2.1.2 Cách thức thao tác XML	13
2.1.2.1 Tạo XML Document	13
2.1.2.2 Xử lý dữ liệu XML	14
2.2 Ngôn ngữ PHP	21
2.2.1 Lịch Sử phát triển	21
2.2.2 Các thành phần cơ bản	22
2.2.3 Xử lý mảng	27
2.2.4 Tương tác tài liệu XML với PHP	27
CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU	31
3.1 Mô tả bài toán	31
3.2 Tổ chức lưu trữ dữ liệu	31
3.2.1 Tập tin lop.xml	32
3.2.2 Tập tin nganhhoc.xml	32
3.2.3 Tập tin sinhvien.xml	33
3.2.4 Tập tin users.xml	36
3.3 Dữ liệu XML qua mô hình DOM	37
3.3.1 Đối với tập tin sinhvien.xml	37
3.3.2 Đối với tập tin lop.xml	38
3.3.3 Đối với tập tin nganhhoc.xml	38
3.3.4 Đối với tập tin users.xml	38
3.3.5 Mô hình khối DOM theo cấu trúc cây	38
3.4 Thiết kế giao diện	40
3.4.1 Giao diện trang đăng nhập	40
3.4.2 Giao diện trang quản lý thông tin sinh viên	40
3.4.3 Giao diện xem danh sách sinh viên	41
3.4.4 Giao diện trang tìm kiếm	41
3.4.5 Giao diện xem lý lịch sinh viên	42
CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU	43
4.1 Dữ liệu thử nghiệm	43
4.2 Kết quả thực nghiệm	45
4.2.1 Chức năng đăng nhập hệ thống	45
4.2.2 Chức năng quản lý thông tin sinh viên	46
4.2.3 Tìm kiếm theo danh sách lớp	48
4.2.4 Tìm kiếm theo danh sách ngành học	49
4.2.5 Cập nhật thông tin cá nhân của người dùng	49
4.2.6 Xem lý lịch trích ngang của bản thân	50
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	51

5.1	Kết luận.....	51
5.2	Hướng phát triển	51
DANH MỤC TÀI LIỆU THAM KHẢO.....		52

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 3.1 Cấu trúc lưu trữ dữ liệu dưới dạng file .xml	31
Hình 3.2 Thiết kế giao diện đăng nhập	40
Hình 3.3 Thiết kế giao diện chính.....	40
Hình 3.4 Thiết kế giao diện xem danh sách sinh viên	41
Hình 3.5 Thiết kế giao diện tìm kiếm sinh viên.....	41
Hình 3.6 Thiết kế giao diện xem lý lịch trích ngang	42
Hình 4.1 Giao diện đăng nhập	45
Hình 4.2 Giao diện người dùng	45
Hình 4.3 Giao diện quản trị	46
Hình 4.4 Giao diện thêm sinh viên	46
Hình 4.5 Giao diện sửa sinh viên.....	47
Hình 4.6 Giao diện xoá sinh viên	48
Hình 4.7 Giao diện xoá sinh viên	48
Hình 4.8 Giao diện tìm kiếm theo lớp	48
Hình 4.9 Giao diện tìm kiếm theo ngành.....	49
Hình 4.10 Giao diện cập nhật thông tin	49
Hình 4.11 Giao diện xem lý lịch trích ngang.....	50
Bảng 3.1 Bảng dữ liệu file lop.xml.....	32
Bảng 3.2 Bảng dữ liệu file nganhhoc.xml	32
Bảng 3.3 Bảng dữ liệu file sinhvien.xml	33
Bảng 3.4 Bảng dữ liệu file users.xml.....	36
Bảng 4.1 Bảng dữ liệu thử nghiệm lớp học	43
Bảng 4.2 Bảng dữ liệu thử nghiệm ngành học.....	43
Bảng 4.3 Bảng dữ liệu thử nghiệm tài khoản người dùng.....	43
Bảng 4.4 Bảng dữ liệu thông tin sinh viên.....	44

TÓM TẮT ĐỒ ÁN THỰC TẬP CƠ SỞ NGÀNH

Vấn đề nghiên cứu:

Ngôn ngữ XML (Extensible Markup Language) là một ngôn ngữ khá mạnh mẽ, thao tác đơn giản được sử dụng rộng rãi trong việc lưu trữ và truyền tải dữ liệu có cấu trúc. Đề tài này tập trung vào việc tìm hiểu ứng dụng ngôn ngữ của XML trong thiết kế cấu trúc lưu trữ dữ liệu sinh viên.

Hướng tiếp cận:

Tìm hiểu XML, nghiên cứu cơ bản về ngôn ngữ XML, cú pháp, và cách dùng để biểu diễn dữ liệu có cấu trúc.

Thiết kế cấu trúc dữ liệu sinh viên, xây dựng một cấu trúc dữ liệu XML cho thông tin sinh viên, bao gồm thông tin cá nhân và tên cha mẹ.

Xử lý dữ liệu XML, tìm hiểu về cách xử lý dữ liệu XML bằng các công nghệ như DOM (Document Object Model).

Ứng dụng thực hành, áp dụng kiến thức để tạo và sửa đổi tệp XML chứa thông tin sinh viên. Kiểm tra quá trình tải tệp và trích xuất thông tin từ XML.

Cách giải quyết vấn đề:

Tìm hiểu ngôn ngữ XML và cách áp dụng nó trong lưu trữ dữ liệu có cấu trúc.

Xây dựng cấu trúc dữ liệu XML phù hợp cho thông tin sinh viên và các thông tin liên quan.

Thực hiện thí nghiệm và ứng dụng thực tế để kiểm tra và xác minh tính hiệu quả của việc lưu trữ thông tin sinh viên dưới dạng XML.

Đồ án sẽ hoàn thiện việc nghiên cứu và áp dụng ngôn ngữ XML vào thiết kế cấu trúc lưu trữ dữ liệu sinh viên, từ đó đem lại cái nhìn tổng quan về khả năng và hiệu quả của XML trong việc này.

Kết quả đạt được: Bản thân có thể làm được một trang website quản lý dữ liệu thông tin sinh viên cơ bản, hiểu rõ về cách tổ chức và quản lý dữ liệu người dùng; Biết được sự tiện lợi và hiệu quả của ngôn ngữ XML trong việc quản lý dữ liệu và cách thao tác với XML.

MỞ ĐẦU

Lý do chọn đề tài

Lựa chọn đề tài “Tìm hiểu ngôn ngữ XML và thiết kế cấu trúc lưu trữ dữ liệu sinh viên minh họa” là kết quả của việc tỉ mỉ tìm hiểu và lựa chọn từ danh sách đề tài mà nhiều giáo viên cung cấp. Trong những đề tài này, tôi nhận thức được rằng có nhiều đề tài phù hợp với tôi nhưng một số còn lại quá khó khăn. Sau cùng tôi quyết định chọn đề tài này, hiện tại là do sự hấp dẫn của việc khám phá ngôn ngữ mới là XML trong việc lưu trữ và truyền tải dữ liệu.

Đặc biệt, sự linh hoạt, mạnh mẽ, dễ tìm hiểu và sử dụng của XML đã làm tôi đánh giá cao ngôn ngữ này. Qua việc nghiên cứu sâu, tôi hiểu rõ cách tạo lập cấu trúc và các tiện ích của ngôn ngữ đánh dấu này. Sự hứng thú này không chỉ giúp tôi mở rộng kiến thức của mình mà còn đưa ra cơ hội để tìm hiểu về ngôn ngữ liên quan như HTML, PHP, CSS, và DOM. Điều này sẽ giúp tôi củng cố kiến thức hiện có và đồng thời đạt được sự đa dạng trong chuyên ngành.

Chính vì vậy, tôi đã lựa chọn thực hiện đề tài “Tìm hiểu ngôn ngữ XML và thiết kế cấu trúc lưu trữ dữ liệu sinh viên minh họa”.

Mục đích chọn đề tài

Việc nghiên cứu đề tài này trước hết giúp tôi tìm hiểu một cách khái quát về XML. Từ việc hiểu cách thức tạo lập cũng như các tiện ích của ngôn ngữ đánh dấu.

Qua việc tìm hiểu trên, tôi thấy khá hứng thú với đề tài này và cũng muốn bản thân có nhiều kiến thức hơn thông qua việc tìm hiểu một ngôn ngữ mà mình chưa biết tới. Liên quan đến đề tài còn có một số ngôn ngữ như HTML, PHP, CSS, DOM. Thông qua đó giúp tôi bổ sung được một lượng kiến thức mới cũng như củng cố lại kiến thức cũ mà mình đã có.

Đối tượng nghiên cứu

Thông tin sinh viên cần lưu trữ.

Ngôn ngữ XML: cách tổ chức lưu trữ dữ liệu.

Tìm hiểu ngôn ngữ PHP và cách tương tác dữ liệu từ tài liệu XML.

Ngôn ngữ HTML, CSS để thiết kế giao diện.

Phạm vi nghiên cứu

Xây dựng một website giới thiệu sản phẩm công nghệ minh họa cách tổ chức lưu trữ dữ liệu với XML. Quá trình này sẽ bao gồm sự tương tác giữa ngôn ngữ XML và ngôn ngữ PHP, cũng như sử dụng HTML và CSS để thiết kế giao diện trực quan. Tầm quan trọng của việc này là giúp tôi áp dụng kiến thức vào thực tế và đồng thời cung cấp một sản phẩm minh họa rõ ràng về cách tổ chức lưu trữ dữ liệu với sự kết hợp linh hoạt của ngôn ngữ XML.

CHƯƠNG 1 TỔNG QUAN

Đề tài “Tìm hiểu ngôn ngữ XML và thiết kế cấu trúc lưu trữ dữ liệu sinh viên minh họa” tập trung vào việc nghiên cứu về ngôn ngữ XML (eXtensible Markup Language) và áp dụng kiến thức đó để thiết kế cấu trúc lưu trữ dữ liệu về sinh viên.

1.1 Tổng quan về vấn đề

Xác định yêu cầu: Đặt ra các yêu cầu cụ thể về thông tin cần lưu trữ về sinh viên bao gồm việc sắp xếp thông tin vào thẻ, xác định cấu trúc cây và quy định cách thức lưu trữ dữ liệu.

Sử dụng thẻ XML để biểu diễn mỗi phần thông tin

Ví Dụ:

`<student>` là thẻ chính trong đó có các thẻ con như `<full_name>`, `<class_code>`, `<gender>`, ...

Sử dụng một vài thuộc tính để mô tả và giảm kích thước cấu trúc XML và làm cho dữ liệu trở nên dễ đọc.

Ví Dụ:

```
<parents>
  <father full_name="Nguyen Van A" occupation="Kỹ
Su" year_of_birth="1980 "/>
  <mother full_name="Nguyen Van B "
occupation="Tester" year_of_birth="1999 "/>
</parents>
```

1.2 Cách giải quyết

Tìm hiểu chi tiết về cú pháp XML, các nguyên tắc cơ bản và cách nó hoạt động, cách tổ chức lưu trữ dữ liệu dưới dạng tập tin.

Xác định rõ các thông tin cụ thể liên quan đến sinh viên mà hệ thống cần lưu trữ. Xây dựng cấu trúc XML dựa trên yêu cầu dữ liệu đã xác định, sử dụng các phần tử và thuộc tính của XML một cách hợp lý để đảm bảo cấu trúc phản ánh đầy đủ thông tin và làm cho nó dễ hiểu và linh hoạt.

Nghiên cứu về cách xử lý dữ liệu XML với ngôn ngữ lập trình PHP. Thực hiện kiểm tra các trường hợp ràng buộc để đảm bảo cấu trúc lưu trữ đủ và đúng quy cách. Đánh giá khả năng mở rộng của cấu trúc để thêm các thuộc tính mới mà không làm thay đổi cấu trúc cơ bản.

Xem xét cách cấu trúc XML có thể tích hợp vào các hệ thống quản lý thông tin sinh viên hiện tại, nghiên cứu về cách XML có thể tương tác với các công nghệ khác như cơ sở dữ liệu hoặc web services.

CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT

2.1 Ngôn ngữ XML

2.1.1 Giới thiệu tổng quan

XML viết tắt của “**Extensible Markup Language**” là ngôn ngữ đánh dấu mở rộng, được thiết kế để tổ chức, lưu trữ và truyền tải dữ liệu có cấu trúc.

Ưu điểm của XML: XML lưu trữ dữ liệu độc lập, được sử dụng để mô tả dữ liệu dưới dạng văn bản, vì thế hầu hết các chương trình hay các phần mềm đều có thể đọc được chúng. XML có thể đọc và phân tích nguồn dữ liệu khá dễ dàng với mục đích trao đổi dữ liệu giữa các chương trình và các hệ thống khác nhau. [1]

Hạn chế của XML: XML cũng có sai sót khi truyền dữ liệu mặc dù khả năng xảy ra không cao.

XML sử dụng thẻ bao quanh các phần tử, các thẻ XML kết hợp theo quy tắc sẽ tạo thành tập tin XML. Các thẻ trong XML phải được đặt trong một thứ tự chính xác, mỗi thẻ có thể chứa các thuộc tính và giá trị.

2.1.1.1 Phần tử gốc và phần tử con

Tài liệu XML bắt đầu bằng một phần tử gốc chứa tất cả các phần tử khác, mỗi phần tử có thể chứa nhiều phần tử con.

Các phần tử XML có thể được định nghĩa là các container để giữ văn bản, phần tử, thuộc tính và đối tượng. Mỗi phần tử XML chứa một hoặc nhiều phần tử, phạm vi của chúng được giới hạn bởi các thẻ đóng và thẻ mở hoặc được giới hạn bởi các phần tử trống. [1]

2.1.1.2 Thuộc tính

Thuộc tính là các thông tin bổ sung được đặt trong thẻ mở của một phần tử XML để mô tả và cung cấp thông tin chi tiết về nội dung của nó.

Cú pháp:

```
<elementName attributeName = "attributeValue">  
Content  
</elementname>
```

Ví dụ:

```
<person name="To Van Toi" age="21">
  <address city="Tra Vinh">huyen Tra Cu</address>
</person>
```

Đặc điểm và quy ước khi đặt thuộc tính

Thuộc tính thường đặt ở thẻ mở của một phần tử XML.

Mỗi thuộc tính phải có một giá trị và giá trị đặt trong dấu ngoặc kép hoặc ngoặc đơn.

Giá trị thuộc tính không thể chứa các kí tự đặc biệt như '<','>','&'.

2.1.1.3 Ví dụ cấu trúc lưu trữ dữ liệu

Đây là ví dụ về cấu trúc trong tập tin **sinhvien.xml**

```
<?xml version = "1.0" encoding = "UTF-8"?>
<students>
  <student>
    <mssv>110121252<mssv>
    <full_name>Tô Văn Tới<full_name>
    <class_code>DA21TTA<class_code>
  </student>
</students>
```

2.1.2 Cách thức thao tác XML

2.1.2.1 Tạo XML Document

Sử dụng trình soạn thảo văn bản hoặc các ngôn ngữ lập trình hỗ trợ và có thể kết hợp thêm các thư viện và API XML trong các ngôn ngữ như Python, Java,C#.

Bước 1: Khai báo phiên bản.

Bước 2: Tạo phần tử gốc.

Bước 3: Tạo các phần tử con

Các qui tắc khi tạo tài liệu XML

- Phải có phần tử gốc (root) duy nhất.
- Mỗi thẻ mở và đóng phải giống nhau về tên.
- Mỗi phần tử con phải nằm trọn bên trong phần tử cha.
- Giá trị thuộc tính phải được đặt giữa cặp nháy đơn hoặc dấu nháy đôi.

2.1.2.2 Xử lý dữ liệu XML

Sử dụng mô hình DOM (Document Object Model)

DOM là một phương pháp được sử dụng để truy cập và thay đổi nội dung của tài liệu HTML và XML. Đối với tài liệu XML, DOM cho phép nhà phát triển tương tác với dữ liệu XML như một cấu trúc cây có tổ chức. [1]

Một số đặc điểm của DOM trong XML:

Cấu trúc cây: DOM biểu diễn dữ liệu XML dưới dạng cây, với mỗi phần tử, thuộc tính, văn bản và các thành phần khác được tổ chức thành các nút trong cây.

Tương tác dữ liệu: DOM cho phép truy cập, thay đổi, thêm hoặc xóa các phần tử trong cây XML.

Đối Tượng Node: Mỗi thành phần trong cây DOM được biểu diễn dưới dạng đối tượng (node). Có nhiều loại node, bao gồm các node kiểu Element (đại diện cho các phần tử XML), các node kiểu Attribute (đại diện cho các thuộc tính), và các node kiểu Text (đại diện cho văn bản).

Ví dụ: Thao tác đọc giá trị của một phần tử XML có thể sử dụng các phương thức như `getElementById()` hoặc `getElementsByTagName()` đối với ngôn ngữ JavaScript

Xử Lý Sự Kiện: DOM cho phép xử lý sự kiện, giúp lập trình viên đáp ứng và xử lý các sự kiện như click, hover và thay đổi dữ liệu.

Ứng dụng trong XML: DOM thường được sử dụng để đọc và biên soạn tài liệu XML, cũng như để tương tác với dữ liệu XML trong các ứng dụng web và môi trường lập trình khác.

SAX (Simple API for XML): Là một giao diện lập trình ứng dụng (**API**) trong ngôn ngữ Java được thiết kế để đọc và xử lý tài liệu XML một cách có hiệu suất cao và bộ nhớ hiệu quả. SAX không phải là một mô hình đối tượng như DOM (**Document Object Model**), thay vào đó nó là một giao diện dựa trên sự kiện (**event-driven**). Điều này có nghĩa là SAX không tải toàn bộ tài liệu XML vào bộ nhớ mà thay vào đó tạo ra các sự kiện khi duyệt qua từng phần tử trong tài liệu. [1]

Một số đặc điểm quan trọng của SAX:

Event-Driven: SAX tạo ra các sự kiện (**events**) khi duyệt qua từng phần tử trong tài liệu XML. Các sự kiện bao gồm bắt đầu một phần tử, kết thúc một phần tử, hoặc gặp dữ liệu văn bản.

Không lưu trữ toàn bộ tài liệu trong bộ nhớ: SAX đọc tài liệu XML theo từng phần một và không cần phải lưu trữ toàn bộ tài liệu trong bộ nhớ, điều này giúp giảm tiêu tốn bộ nhớ khi xử lý các tài liệu lớn.

Khả năng xử lý lớn: SAX là lựa chọn phổ biến cho việc xử lý các tài liệu XML lớn hoặc dữ liệu liên tục, nơi việc lưu trữ toàn bộ tài liệu trong bộ nhớ có thể trở thành vấn đề.

SAX chỉ hỗ trợ việc đọc (**read-only**) tài liệu XML. Nó không cung cấp khả năng chỉnh sửa tài liệu.

Phương Thức Callback: Các ứng dụng sử dụng SAX thông qua việc triển khai các phương thức gọi lại (**callback**) để xử lý sự kiện khi chúng xảy ra.

Ví dụ: Cách sử dụng SAX

```
class MySAXHandler extends DefaultHandler {  
  
    public function startElement($name, $attrs) {  
  
        // Xử lý khi bắt đầu một phần tử (element)  
        echo "Bắt đầu phần tử: $name\n";  
  
        // Hiển thị các thuộc tính của phần tử  
        foreach ($attrs as $attribute => $value) {  
  
            echo "Thuộc tính: $attribute =  
$value\n";  
        }  
    }  
}
```

```
    }

    }

    public function characters($data) {

        // Xử lý dữ liệu trong các phần tử

        echo "Dữ liệu: $data\n";

    }

    public function endElement($name) {

        // Xử lý khi kết thúc một phần tử

        echo "Kết thúc phần tử: $name\n";

    }

}

// Tạo một đối tượng phân tích SAX
$xmlParser = xml_parser_create();
$xmlHandler = new MySAXHandler();

// Thiết lập các hàm xử lý cho phân tích SAX
xml_set_object($xmlParser, $xmlHandler);
xml_set_element_handler($xmlParser, "startElement",
"endElement");

xml_set_character_data_handler($xmlParser,
"characters");

// Đọc tệp XML

$xmlFile = "duongdan/tep.xml"; // Thay đổi đường
dẫn tới tệp XML cần phân tích

$file = fopen($xmlFile, "r");

while ($data = fread($file, 4096)) {

    xml_parse($xmlParser, $data, feof($file)) or
```

```
die(sprintf("Lỗi phân tích XML: %s tại dòng
%d",
xml_error_string(xml_get_error_code($xmlParser)),
xml_get_current_line_number($xmlParser)));
}

// Giải phóng bộ nhớ và đóng tệp XML
xml_parser_free($xmlParser);

fclose($file);
```

XPath (**XML Path Language**) là một ngôn ngữ truy vấn được thiết kế để định vị và trích xuất dữ liệu từ tài liệu XML. Nó là một phần quan trọng của các công nghệ liên quan đến XML như XSLT (**Extensible Stylesheet Language Transformations**) và XQuery. XPath cho phép bạn duyệt qua cấu trúc phức tạp của một tài liệu XML để lựa chọn và trích xuất các phần tử, thuộc tính và nội dung khác một cách linh hoạt. [1]

Một số khái niệm cơ bản trong XPath:

Biểu thức Đường dẫn (Path Expression): XPath sử dụng biểu thức đường dẫn để duyệt qua các phần tử và thuộc tính trong tài liệu XML.

Ví dụ:

```
/bookstore/book/title
```

Trong biểu thức này, chúng ta đang lựa chọn tất cả các phần tử <title> nằm trong phần tử <book> của phần tử <bookstore>.

Thuộc Tính và Giá Trị: XPath cho phép lựa chọn dữ liệu dựa trên giá trị của thuộc tính.

Ví dụ:

```
//book[@category='fiction']
```

Biểu thức này lựa chọn tất cả các phần tử <book> có thuộc tính category có giá trị là 'fiction'.

Wildcards (Ký tự đại diện): XPath hỗ trợ ký tự đại diện như * và @*, cho phép lựa chọn tất cả hoặc một số phần tử/thuộc tính mà không cần chỉ định tên cụ thể.

Ví dụ:

```
//book/*
```

Lựa chọn tất cả các phần tử con của <book>.

Axes (Trục): XPath có các trục giúp duyệt qua cấu trúc tài liệu theo các hướng khác nhau.

Ví dụ:

```
//book/following-sibling::book
```

Lựa chọn tất cả các phần tử <book> là anh em sau phần tử <book> được chọn trước đó.

Hàm XPath: XPath cung cấp một số hàm tích hợp để thực hiện các thao tác như đếm, lấy giá trị tối đa/tối thiểu, và so sánh.

Ví dụ:

```
//book[position()=1]
```

Lựa chọn phần tử <book> đầu tiên trong tài liệu.

XPath là một công cụ mạnh mẽ cho việc truy cập và trích xuất dữ liệu từ tài liệu XML và nó được sử dụng rộng rãi trong các ngữ cảnh như truy vấn XML, biến đổi XML, và kiểm thử XML.

XSLT (Extensible Stylesheet Language Transformations)

XSLT là một ngôn ngữ được sử dụng để chuyển đổi và biến đổi dữ liệu XML thành định dạng khác. XSLT hoạt động như một ngôn ngữ định kiểu Stylesheet, cho phép định nghĩa các quy tắc và mẫu để ánh xạ một tài liệu XML sang một cấu trúc hoặc định dạng mới [1].

Một số đặc điểm của XSLT:

Định Nghĩa Stylesheet: Một stylesheet XSLT thường được viết dưới dạng tài liệu XML. Nó chứa các quy tắc và mẫu mô tả cách dữ liệu XML nên được chuyển đổi.

Templates và Patterns: XSLT sử dụng mô hình template và pattern để ánh xạ các phần của tài liệu nguồn sang định dạng đích. Một template là một mẫu mô tả cách một phần cụ thể của tài liệu nguồn nên được xử lý.

Biến đổi dữ liệu: XSLT có khả năng biến đổi cả cấu trúc và nội dung của tài liệu XML. Các mẫu và quy tắc được áp dụng để chọn và xử lý các phần tử, thuộc tính và văn bản trong tài liệu nguồn.

Hỗ trợ nhiều định dạng: XSLT có thể tạo ra đầu ra ở nhiều định dạng khác nhau bao gồm XML, HTML, văn bản thuần, hoặc bất kỳ định dạng nào khác.

XSLT thường được sử dụng trong trình duyệt web để biến đổi và hiển thị dữ liệu XML mà không cần tải lại trang. Nó thường kết hợp với XML và XSLT được áp dụng trực tiếp trong trình duyệt để tạo ra giao diện người dùng động.

Tính mở rộng và tích hợp: XSLT có thể được mở rộng thông qua sử dụng các phần mở rộng và chức năng mạnh mẽ như biến, hàm và xử lý lỗi. Nó có thể tích hợp với các ngôn ngữ lập trình khác thông qua các giao diện và API.

XSLT là một công cụ mạnh mẽ, cung cấp khả năng biến đổi linh hoạt và hiệu quả cho dữ liệu XML, làm cho nó trở thành một phần quan trọng trong xử lý và biểu diễn dữ liệu trên web và trong các ứng dụng khác.

Xquery

XQuery là một ngôn ngữ truy vấn (query language) được thiết kế đặc biệt để truy vấn và xử lý dữ liệu XML. Được đề xuất bởi W3C (World Wide Web Consortium), XQuery cung cấp một cách mạnh mẽ để trích xuất thông tin từ tài liệu XML, thực hiện các phép biến đổi dữ liệu, và tạo ra kết quả theo định dạng XML.

Một số đặc điểm của XQuery:

XQuery sử dụng cú pháp tương tự như SQL, nhưng được thiết kế đặc biệt để làm việc với dữ liệu XML. Câu truy vấn XQuery thường có cấu trúc lồng nhau và sử dụng các biểu thức XPath để chỉ định vị trí của dữ liệu trong tài liệu XML.

Trích xuất và lọc dữ liệu: XQuery cho phép trích xuất thông tin từ các phần tử, thuộc tính hoặc giá trị cụ thể trong tài liệu XML. Các điều kiện có thể được áp dụng để lọc dữ liệu theo một tập hợp các điều kiện.

Biểu Thức XPath: XQuery sử dụng biểu thức XPath để định vị vị trí cụ thể trong tài liệu XML. Điều này cho phép người dùng chỉ định nơi mà truy vấn sẽ được thực hiện và lấy dữ liệu từ các vị trí đó.

XQuery là một tiêu chuẩn của W3C, điều này đảm bảo tính di động và tính tương thích giữa các hệ thống hỗ trợ XQuery khác nhau. Ngoài việc truy vấn dữ liệu, XQuery cũng hỗ trợ việc biến đổi dữ liệu bằng cách thêm, xóa, hoặc cập nhật các phần tử trong tài liệu XML. XQuery có thể tích hợp chặt chẽ với XML Schema giúp đảm bảo tính nhất quán và kiểm soát về mặt cấu trúc của dữ liệu XML.

XSD(XML Schema Definition) là một ngôn ngữ mô tả cấu trúc và các ràng buộc của tài liệu XML. Nó định nghĩa các quy tắc và kiểu dữ liệu mà tài liệu XML cần tuân thủ, cung cấp một phương tiện mạnh mẽ để xác định và kiểm soát cấu trúc của dữ liệu XML.

Một số đặc điểm của XSD:

XSD được thiết kế để định nghĩa cấu trúc của tài liệu XML. Nó cho phép mô tả chi tiết về các phần tử, thuộc tính, kiểu dữ liệu và ràng buộc trong tài liệu XML bao gồm `<element>`, `<complexType>`, `<simpleType>`, `<attribute>`.

Phần Tử `<element>`: định nghĩa một phần tử trong tài liệu XML và xác định kiểu dữ liệu của nó. Nó có thể mô tả phần tử là phần tử đơn lẻ, phần tử chứa các phần tử khác, hoặc phần tử chứa văn bản.

Phần Tử `<complexType>` và `<simpleType>`: được sử dụng để xác định kiểu dữ liệu của một phần tử.

`<complexType>` cho phép định nghĩa cấu trúc phức tạp hơn bằng cách kết hợp các phần tử và thuộc tính.

`<simpleType>` làm cho quá trình xác định kiểu dữ liệu đơn giản hơn.

Phần Tử `<attribute>`: định nghĩa một thuộc tính của một phần tử và xác định kiểu dữ liệu của thuộc tính đó.

XSD cung cấp một loạt các kiểu dữ liệu cơ bản như string, integer, boolean, date, time, và cho phép người định nghĩa cấu trúc kiểu dữ liệu tùy chỉnh.

XSD cho phép áp dụng ràng buộc lên dữ liệu, ví dụ như ràng buộc giới hạn giá trị tối thiểu và tối đa, ràng buộc định dạng và ràng buộc khác.

XSD thường được sử dụng như một tài liệu riêng biệt hoặc nhúng trực tiếp vào tài liệu XML thông qua một phân tử `<schema>`.

Sử dụng XSD giúp đảm bảo tính đúng đắn và nhất quán của dữ liệu XML, đồng thời cung cấp tài liệu rõ ràng và chi tiết trong các ứng dụng và hệ thống.

AJAX (Asynchronous JavaScript and XML) là một kỹ thuật lập trình web cho phép gửi và nhận dữ liệu từ máy chủ mà không cần nạp lại toàn bộ trang web.

Một số đặc điểm của AJAX và mối liên quan của nó với XML:

Gửi và nhận dữ liệu bất đồng bộ: AJAX cho phép trang web gửi và nhận dữ liệu từ máy chủ mà không cần nạp lại trang web. Điều này giúp tăng trải nghiệm người dùng và làm giảm độ trễ.

XMLHttpRequest: Để thực hiện gửi và nhận dữ liệu không đồng bộ, AJAX sử dụng đối tượng XMLHttpRequest trong JavaScript. Đối tượng này làm cho việc gửi và nhận dữ liệu trở nên dễ dàng và hiệu quả.

Sử dụng với XML hoặc JSON: Ban đầu, AJAX thường được sử dụng để tương tác với dữ liệu XML. Tuy nhiên, do tính linh hoạt, nó cũng được sử dụng rộng rãi với định dạng dữ liệu JSON, đặc biệt là khi làm việc với các API web hiện đại.

Các phương thức HTTP: AJAX sử dụng các phương thức HTTP như GET và POST để gửi và nhận dữ liệu từ máy chủ.

2.2 Ngôn ngữ PHP

PHP viết tắt của “Hypertext Preprocessor” là một ngôn ngữ lập trình mã nguồn mở được thiết kế đặc biệt cho việc phát triển ứng dụng web, PHP chủ yếu thực hiện các tác vụ trên máy chủ và tạo ra nội dung động trên trang web. [2]

2.2.1 Lịch Sử phát triển

PHP được tạo ra lần đầu tiên bởi Rasmus Lerdorf vào tháng 12 năm 1994. Ban đầu, nó chỉ là một bộ mã CGI (**Common Gateway Interface**) để theo dõi ngày và thời gian khi Rasmus duyệt web.

PHP/FI: Năm 1995, Rasmus Lerdorf phát triển PHP/FI (**Personal Home Page/Forms Interpreter**), phiên bản đầu tiên của PHP. Nó đã giúp các nhà phát triển tạo các trang web động và xử lý các biểu mẫu web. [3]

PHP 3: Trong giai đoạn tiếp theo, vào tháng 6 năm 1998, PHP 3 đã được ra mắt. Lần này, Zeev Suraski và Andi Gutmans, hai nhà phát triển, đã thêm vào mã nguồn ngôn ngữ một trình thực thi mới, Zend Engine, giúp tăng cường hiệu suất và tính module của PHP. [3]

PHP 4: Được phát hành vào tháng 5 năm 2000, mang lại nhiều tính năng mới như hỗ trợ đối tượng (**OOP**), thư viện xử lý XML, và các cải tiến khác về hiệu suất. [3]

PHP 5: Ra đời vào tháng 7 năm 2004, mang theo Zend Engine 2.0 và một loạt các tính năng mới, bao gồm hỗ trợ đầy đủ cho OOP, xử lý XML cải tiến, và nhiều cải tiến khác. [3]

PHP 7: Được phát hành vào tháng 12 năm 2015 với sự cải tiến về hiệu suất đáng kể. Nó có thể chạy nhanh hơn đến 2 lần so với PHP 5.x và cung cấp nhiều cải tiến khác như khai báo kiểu, hằng số trình biên dịch, và cải thiện cú pháp. [3]

PHP 8: Được ra mắt vào tháng 11 năm 2020, đưa ra nhiều tính năng mới và cải tiến. Một số điểm đáng chú ý bao gồm Just-In-Time Compilation (**JIT**), cải thiện hiệu suất, thêm các tính năng mới trong OOP, và nhiều sự cải tiến khác. [3]

PHP ngày nay không chỉ được sử dụng rộng rãi trong phát triển web mà còn là một phần quan trọng của nhiều dự án và hệ thống lớn trên toàn thế giới. Sự phát triển liên tục và cải thiện của PHP đã giúp nó duy trì sức hấp dẫn trong cộng đồng phát triển web.

Cấu Trúc PHP

2.2.2 Các thành phần cơ bản

Mã PHP được đặt trong thẻ `<?php ... ?>`. [4]

Ví dụ: In câu chào “Hello, World”

```
<?php echo "Hello, World"; ?>
```


2.2.2.1 Biến

Biến trong PHP là một đối tượng dùng để lưu trữ thông tin hoặc giá trị. Mỗi biến trong PHP có một tên duy nhất và chứa dữ liệu của một kiểu cụ thể và tên biến bắt đầu bằng ký hiệu \$. [4]

Cú pháp:

```
$tên_biến = "giá trị của biến" ;
```

Ví dụ:

```
$name = "To Van Toi" ;
```

Quy tắc đặt tên biến: Tên biến có thể bắt đầu bằng một chữ cái hoặc dấu gạch dưới '_', sau đó có thể sử dụng chữ cái, số hoặc dấu gạch dưới trong tên biến. Biến trong PHP phân biệt chữ hoa và chữ thường.

2.2.2.2 Hàm

Trong PHP, hàm là một khối mã chức năng có thể tái sử dụng, được đặt tên và thực hiện một nhiệm vụ cụ thể. Việc sử dụng hàm giúp chia nhỏ chương trình thành các phần nhỏ hơn, dễ quản lý và tái sử dụng. [4]

Cú pháp:

```
function tên_hàm($tham_so1, $tham_so2,...) {  
    // mã chức năng  
}
```

Ví dụ:

```
<?php  
  
    function ham() {      echo 1 + 2;    }  
  
?>
```

Gọi hàm: Sử dụng tên của nó và truyền các đối số theo thứ tự được định nghĩa trong hàm.

Ví dụ:

```
<?php
```

```
function ham() { echo 1 + 2; }

ham();

?>
```

2.2.2.3 Biểu Thức

Biểu thức là một tổ hợp các toán hạng và toán tử. Toán tử thực hiện các thao tác như cộng, trừ, nhân, chia, so sánh, ... Toán hạng là những biến hay là những giá trị mà các phép toán được thực hiện trên nó. Ví dụ $a + b$ thì a và b được gọi là toán hạng, dấu $+$ được gọi là toán tử, cả 2 kết hợp lại thành một biểu thức $(a + b)$.

Mỗi biểu thức chỉ có một giá trị nhất định. Ví dụ ta có biểu thức $(a + b)$ thì biểu thức này có giá trị là tổng của a và b . [4]

Ví dụ:

```
$ketqua = $a - $b;

$ketqua = 7 + 6;

$ketqua = 3*$x + 4*$y;
```

Toán tử gán (Assignment Operator): Đây là toán tử thông dụng nhất trong mọi ngôn ngữ, ta dùng dấu '=' để gán giá trị cho một biến bất kỳ nào đó.

Ví dụ:

```
$a = 12;
```

Nhiều biến có thể được gán cùng một giá trị qua một câu lệnh đơn gọi là gán liên tiếp.

Ví dụ:

```
$a = $b = $c = $d = 12;
```

Biểu thức số học: Các phép toán thường được thực hiện theo một thứ tự cụ thể gọi là độ ưu tiên để đưa ra giá trị cuối cùng. Các biểu thức số học trong các ngôn ngữ được thể hiện bằng cách sử dụng các toán tử số học cùng với các toán hạng dạng số hoặc ký tự (**biến**). Những biểu thức này gọi là biểu thức số học.

Ví dụ:

```
$ketqua = $a + $b/2;  
  
$ketqua = $a / 7;  
  
$ketqua = $a + ($b = 5 + 6);
```

Như ta thấy trên toán hạng có thể là hằng, biến hay kết hợp cả 2 và mỗi biểu thức có thể kết hợp của nhiều biểu thức con. Định nghĩa hằng và biến như thế nào chúng ta đã đề cập ở Biến Và Hằng Số Trong PHP.

Toán tử quan hệ: Được dùng để kiểm tra mối quan hệ giữa hai biến hay giữa một biến và một hằng số. Ví dụ kiểm tra 2 biến \$a và \$b xem biến nào lớn hơn ta làm như sau: (\$a > \$b) và kết quả của biểu thức này sẽ trả về kiểu boolean **TRUE** hoặc **FALSE**.

Ví dụ:

```
$a = 12; // Biến $a kiểu INT có giá trị = 12  
  
$t = ($a == 12); // Biến $t có giá trị là TRUE vì  
biểu thức (12 == 12) đúng  
  
$t = ($a > 12); // Biến $t có giá trị là FALSE vì  
biểu thức (12 > 12) sai  
  
$t = ($a >= 12); // Biến $t có giá trị TRUE vì  
biểu thức (12 >= 12) đúng  
  
$t = ($a != 12); // Biến $t có giá trị FALSE vì  
biểu thức (12 != 12) sai
```

Toán tử luận lý: Là ký hiệu dùng để kết hợp hay phủ định biểu thức có chứa các toán tử quan hệ, những biểu thức dùng toán tử luận lý trả về giá trị **TRUE** hoặc **FALSE**.

Ví dụ:

```
$a = 100; $b = 200; $tong = $a + $b; $check = ($a <  
$b) && ($tong > 200);
```

2.2.2.4 Cấu trúc điều khiển

PHP cung cấp các cấu trúc điều khiển như **if**, **else**, **for**, **while** để quản lý luồng điều khiển của chương trình.

Ví dụ:

```
<?php

$num = 10 ;

if ($num > 0) {

    echo "positive number" ;

} else {

    echo "non - positive number" ;

}

?>
```

Cấu trúc điều khiển Switch Case trong PHP

Switch...case là một cấu trúc rẽ nhánh, chỉ thực thi một khối lệnh duy nhất tùy thuộc vào giá trị của điều kiện. Nếu không có điều kiện nào được đáp ứng thì khối lệnh trong nhãn default sẽ được thực thi.

Cú pháp:

```
<?php

switch(condition) {

    case value1:

        //block of code to be executed

        break;

    case value2:

        //block of code to be executed

        break;

    default:

        //default block code

}
```

```
        break;

    }

?>
```

2.2.3 Xử lý mảng

Mảng là một cấu trúc dữ liệu lưu giữ một hoặc nhiều kiểu giá trị giống nhau trong một biến. Để tạo mảng chúng ta sử dụng hàm `array()` trong PHP .

Cú pháp:

```
array(

    key  => value,

    key2 => value2,

    key3 => value3,

    ...

)
```

Ví dụ:

```
<?php

    $cars=array("Volvo","BMW","Toyota");

    var_dump( $cars );

?>
```

Mỗi giá trị bên trong mảng sẽ hiển thị kèm theo tên kiểu dữ liệu. Như vậy Volvo, BMW, Toyota là giá trị trong mảng `$cars`. Các key nó sẽ tự động gán khi khởi tạo mảng

Ví dụ:

```
<?php    echo $cars[2]; ?>
```

2.2.4 Tương tác tài liệu XML với PHP

2.2.4.1 Cách đọc dữ liệu từ tài liệu XML

sử dụng thư viện SimpleXML

Đây là một thư viện có sẵn trong PHP để đọc dữ liệu file XML. Thư viện này cung cấp cách tiếp cận thuận tiện để đọc, chỉnh sửa và truy cập dữ liệu từ tệp XML. Thư viện này còn cung cấp một giao diện đơn giản và dễ hiểu cho việc xử lý XML trong PHP.

Ví dụ:

```
$xml = simplexml_load_file('data.xml');
```

Sử dụng lớp DOMDocument

DOMDocument (**Document Object Model**) là một class trong PHP. Class này cho phép tạo, sửa đổi và truy cập vào tài liệu XML dưới dạng cấu trúc cây, cung cấp các phương thức và thuộc tính để thực hiện thao tác trên tài liệu.

Ví dụ:

```
$xml = new DOMDocument();  
$xml->load('sinhvien.xml');
```

2.2.4.2 Cách ghi dữ liệu vào tài liệu XML

Để ghi dữ liệu vào tài liệu XML tôi đã sử dụng các phương thức từ thư viện **SimpleXMLElement** và lớp **DOMDocument**.

Ghi dữ liệu từ SimpleXMLElement

Sử dụng thư viện **SimpleXML** để đọc dữ liệu từ tài liệu XML, sau đó sẽ tạo phần tử mới và gán cho chúng một biến. Tiếp theo sử dụng phương thức **addChild** để thêm phần tử con cho biến đã tạo và **addAttribute** để thêm thuộc tính cho biến. Sau khi đã hoàn thành ta dùng **asXML** để lưu tài liệu XML đã thay đổi vào tệp.

Ví dụ:

```
$xml = simplexml_load_file('users.xml');  
$user = $xml->addChild('user');  
$user->addAttribute('id', $id); // thêm thuộc tính mới  
$user->addChild('mssv', htmlspecialchars($mssv));  
// thêm phần tử mới
```

```
$user->addChild('fullname',
htmlspecialchars($fullname));

$user->addChild('username',
htmlspecialchars($username));

$user->addChild('password',
htmlspecialchars($password));

$user->addChild('phone',
htmlspecialchars($phone));

$user->addChild('email',
htmlspecialchars($email));

$user->addChild('role',
htmlspecialchars($role));

$xml->asXML('users.xml'); // lưu dữ liệu đã thay
đổi vào tệp users.xml
```

Ghi dữ liệu từ DOMDocument

Sử dụng lớp **DOMDocument** dùng phương thức **load()** để tải tài liệu XML, sau đó tạo các phần tử mới từ phương thức **createElement()** và thêm phần tử mới vào tài liệu từ phương thức **appendChild()**. Khi đã hoàn thành ta sẽ lưu dữ liệu vào tệp XML bằng cách sử dụng **save()**.

Ví dụ:

```
$xml = new DOMDocument();

$xml->load('lop.xml');

// Tạo phần tử 'class'
$new_class = $xml->createElement('class');

// Tạo phần tử 'class_id' và 'name' và gán giá
trị từ form
```

```
$class_id_element = $xml->createElement('class_id', $class_id);

$class_name_element = $xml->createElement('name', $class_name);

// Thêm các phần tử vào phần tử 'class'

$new_class->appendChild($class_id_element);

$new_class->appendChild($class_name_element);

// Thêm phần tử 'class' vào phần tử 'classes'
trong tệp XML

$classes = $xml->getElementsByTagName('classes')->item(0);

$classes->appendChild($new_class);

$xml->save('lop.xml'); ` // Lưu tệp XML sau
khi thêm
```


3.1 Mô tả bài toán

Trước hết, tôi mô tả bài toán cụ thể mà nghiên cứu tôi nhằm giải quyết. Bài toán là thiết kế một hệ thống lưu trữ sinh viên sử dụng ngôn ngữ XML. Mục tiêu của tôi là tìm hiểu sâu rộng về khả năng của XML trong việc lưu trữ thông tin sinh viên và xây dựng cấu trúc linh hoạt và hiệu quả.

Trước khi bắt đầu hiện thực hóa, tôi đã phân tích nhu cầu đầy đủ của một hệ thống, bao gồm:

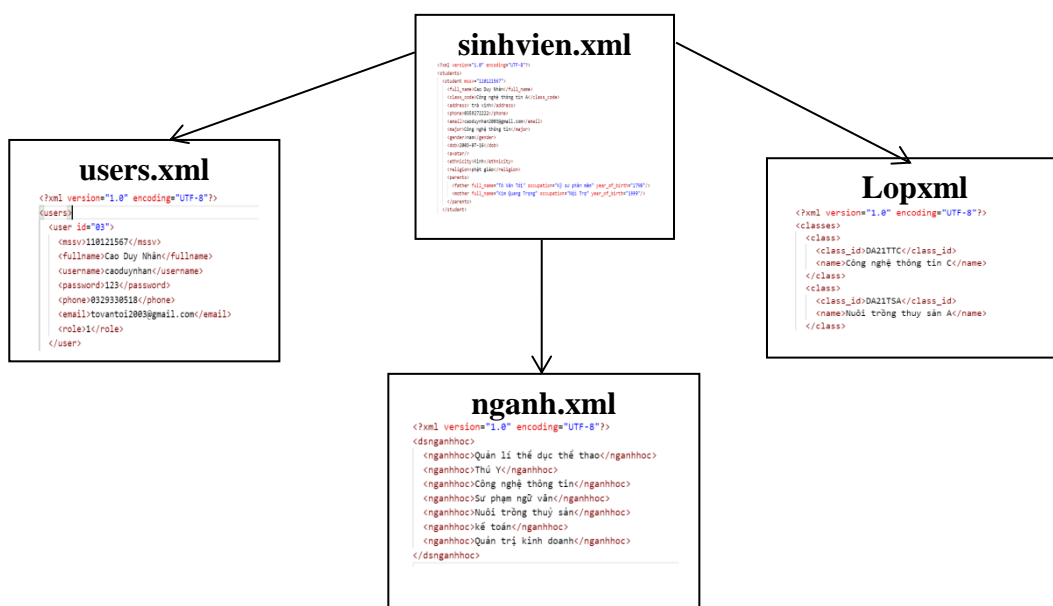
Quản lý thông tin của sinh viên: Hệ thống cần khả năng thêm mới, chỉnh sửa và xóa thông tin về sinh viên.

Tra cứu sinh viên: Người dùng cần có khả năng tra cứu thông tin sinh viên dựa trên các tiêu chí như mã số sinh viên, mã lớp hoặc họ tên.

3.2 Tổ chức lưu trữ dữ liệu

Dữ liệu lưu trữ dưới dạng tập tin .xml và có cấu trúc gồm một phần tử chính là “students” và mỗi phần tử con <student> chứa các thuộc tính như: mssv, full name, class_code ...

Hệ thống lưu trữ dữ liệu dạng tập tin với loại tài liệu là ngôn ngữ đánh dấu XML bao gồm các XML Document:



Hình 3.1 Cấu trúc lưu trữ dữ liệu dưới dạng file .xml

3.2.1 Tập tin lop.xml

Đây là tập XML chứa thông tin về các lớp học trong các ngành. Nó có một phần tử gốc là **<classes>**, trong đó có nhiều phần tử con là **<class>** đại diện cho từng lớp học cụ thể. Mỗi phần tử **<class>** chứa hai thẻ con là **<class_id>** và **<name>**. Trong đó **<class_id>** là mã số của lớp học, **<name>** là tên của lớp học.

Bảng 3.1 Bảng dữ liệu file *lop.xml*

STT	Nội dung	Diễn giải	Ghi chú
1	<classes>		Phần gốc của tập chứa thông tin về các lớp học.
2	<class>		Lưu trữ một lớp học
3	<class_id>	Mã lớp	Mã định danh duy nhất của lớp học.
4	name	Tên lớp	

Ví dụ: Thông tin của một lớp

```
<class>
  <class_id>DA21TTC</class_id>
  <name>Công nghệ thông tin C</name>
</class>
```

3.2.2 Tập tin nganhhoc.xml

Tập tin chứa các thẻ **<nganhhoc>**, mỗi thẻ **<nganhhoc>** đại diện cho một ngành học cụ thể. XML cho phép định nghĩa các thẻ tùy ý, trong trường hợp này là **<nganhhoc>** để chứa thông tin. Các thẻ này có thể được sử dụng để tổ chức và lưu trữ dữ liệu liên quan đến các ngành học khác nhau.

Bảng 3.2 Bảng dữ liệu file *nganhhoc.xml*

STT	Nội dung	Diễn giải	Ghi chú
1	<dsnganhhoc>		Phần gốc của tập chứa thông tin về các ngành học.
2	<nganhhoc>		Lưu trữ thông tin về tên một ngành học
3			
4			

Ví dụ: Thông tin các ngành học

```
<dsnganhhoc>

  <nganhhoc>Thú Y</nganhhoc>

  <nganhhoc>Công nghệ thông tin</nganhhoc>

  <nganhhoc>Su phạm ngữ văn</nganhhoc>

  <nganhhoc>Nuôi trồng thủy sản</nganhhoc>

  <nganhhoc>kế toán</nganhhoc>

  <nganhhoc>Quản trị kinh doanh</nganhhoc>

</dsnganhhoc>
```

3.2.3 Tập tin sinhvien.xml

Phần tử gốc sẽ là 'students' chứa tất cả sinh viên.

Mỗi sinh viên được biểu diễn bằng phần tử 'student'.

Mỗi thông tin của sinh viên là một phần tử con của 'student' như (mssv, full_name, class_code,...).

Phần tử 'parents' chứa thông tin về phụ huynh, với các phần tử con 'father' và 'mother'.

Các thông tin khác như 'ethnicity' và 'religion' nằm trực tiếp trong phần tử 'student'.

Mô tả: Tập XML này chứa thông tin về một sinh viên cụ thể. Thông tin sinh viên được lưu trong các thẻ XML có tên tương ứng với từng trường thông tin.

Bảng 3.3 Bảng dữ liệu file sinhvien.xml

STT	Nội dung	Diễn giải	Ghi chú
1	<students> </students>	Phần tử gốc chứa toàn bộ thông tin về sinh viên trong đoạn mã	
2	<student ></student>	Phần tử con của phần tử gốc đóng vai	

		trò chứa thông tin của một sinh viên cụ thể	
3	mssv	Một số định danh duy nhất được sử dụng để nhận diện và phân biệt giữa các sinh viên. Mỗi sinh viên sẽ có một mssv duy nhất và không trùng lặp.	Attributes
4	<full_name></full_name>	Phần tử đại diện cho thông tin về tên đầy đủ của sinh viên.	
5	<class_code></class_code>	Phần tử chứa mã lớp của sinh viên.	
6	<address></address>	Phần tử chứa địa chỉ của sinh viên.	
7	<phone></phone>	Phần tử chứa số điện thoại của sinh viên.	
8	<email></email>	Phần tử chứa địa chỉ email của sinh viên.	
9	<major></major>	Phần tử chứa thông tin về ngành học của sinh viên.	
10	<gender></gender>	Phần tử chứa giới tính của sinh viên.	
11	<dob></dob>	Phần tử chứa ngày tháng năm sinh của sinh viên.	
12	<avatar></avatar>	Phần tử chứa ảnh đại diện của sinh viên	
13	<ethnicity></ ethnicity>	Phần tử chứa thông tin về dân tộc của sinh viên.	

14	<religion></ religion>	Phần tử chứa thông tin về tôn giáo của sinh viên.	
15	<parents></ parents>	Phần tử chứa thông tin về phụ huynh của sinh viên.	
16	<father></ father>	Phần tử chứa thông tin về cha của sinh viên.	
17	Full_name	Thuộc tính chứa tên đầy đủ của cha sinh viên	Attributes
18	Occupation	Thuộc tính chứa nghề nghiệp của cha sinh viên.	Attributes
19	Year_of_birth	Thuộc tính chứa năm sinh của cha sinh viên.	Attributes
20	<mother></mother>	Phần tử chứa thông tin của mẹ sinh viên	
21	Full_name	Thuộc tính chứa tên đầy đủ của mẹ sinh viên	Attributes
22	Occupation	Thuộc tính chứa nghề nghiệp của mẹ sinh viên.	Attributes
23	Year_of_birth	Thuộc tính chứa năm sinh của mẹ sinh viên.	Attributes

Ví dụ: Thông tin một sinh viên

```
<student mssv="110121253">

    <full_name>Kim Hoàng Nam</full_name>

    <class_code>Công nghệ thông tin C</class_code>

    <address>trà cú - trà vinh</address>

    <phone>0359272288</phone>
```

```

<email>kimhoangnam2003@gmail.com</email>

<major>Công nghệ thông tin</major>

<gender>nam</gender>

<dob>2003-06-05</dob>

<avatar>images (3).jpg</avatar>

<ethnicity>nu</ethnicity>

<religion>phật giáo</religion>

<parents>

    <father full_name="Tô Văn Tới" occupation="Kỹ
su phần mềm" year_of_birth="1798"/>

    <mother full_name="Ngô Thị Trúc Xuân"
occupation="Nội Trợ" year_of_birth="1999"/>

</parents>

</student>

```

3.2.4 Tập tin users.xml

Tập tin có mỗi thẻ <user> lưu trữ thông tin của một người dùng. Trong đó gồm một thuộc tính xác định mã định danh của người dùng và tập hợp các thẻ lưu thông tin của người dùng bao gồm mã số sinh viên, họ tên, tên đăng nhập, mật khẩu, số điện thoại và địa chỉ email.

Bảng 3.4 Bảng dữ liệu file users.xml

STT	Nội dung	Diễn giải	Ghi chú
1	<users>	Phần gốc của tệp chứa thông tin về các ngành học.	
2	<user>	Lưu thông tin một người dùng	
3	id	Lưu mã người dùng	Attribute
4	<mssv></mssv>	Lưu mã số sinh viên	
5	<fullname></fullname>	Lưu họ tên đầy đủ	
6	<password></password>	Lưu mật khẩu	

7	<phone></phone>	Lưu số điện thoại	
8	<email></email>	Lưu địa chỉ email	
9	<role></role>	Vai trò người dùng	

Ví dụ: Thông tin một người dùng

```
<user id="02">
    <mssv>110121252</mssv>
    <fullname>Tô Văn Tới</fullname>
    <username>tovantoi</username>
    <password>123</password>
    <phone>+84359272229</phone>
    <email>tovantoi2003@gmail.com</email>
    <role>1</role>
</user>
```

3.3 Dữ liệu XML qua mô hình DOM

Tôi sử dụng mô hình DOM trong việc tương tác với dữ liệu XML, để có thể dễ dàng thao tác trong việc thêm xóa sửa các chức năng của hệ thống và truy xuất thông tin thông qua các phương thức như tạo một DOM Document, load dữ liệu từ một tệp đã tồn tại, truy cập và thao tác dựa trên cấu trúc cây và lưu dữ liệu.

3.3.1 Đối với tập tin sinhvien.xml

Đọc Tất Cả Sinh Viên: Phương thức để truy xuất danh sách tất cả sinh viên từ tệp XML. Cung cấp thông tin về tất cả các trường thông tin cho mỗi sinh viên.

Truy Cập Thông Tin Sinh Viên Cụ Thể: Có thể truy cập thông tin chi tiết của một sinh viên thông qua mã số sinh viên (mssv).

Trả về thông tin như họ tên, mssv, lớp, địa chỉ, số điện thoại, email, ngành học, giới tính, ngày tháng năm sinh, hình ảnh đại diện, dân tộc, tôn giáo và thông tin về phụ huynh.

3.3.1.1 Thêm, Xóa, Sửa Dữ Liệu

Thêm Sinh Viên Mới: Cung cấp chức năng để thêm một sinh viên mới vào tệp XML với các thông tin tương ứng.

Xóa Sinh Viên: Phương thức để xóa thông tin của một sinh viên dựa trên mã số sinh viên .

Cập Nhật Thông Tin Sinh Viên: Cho phép cập nhật thông tin của một sinh viên đã có trong tệp XML. Có thể cập nhật toàn bộ hoặc một số trường thông tin cụ thể.

3.3.2 Đối với tệp tin lop.xml

Đọc Dữ Liệu: Phương thức để đọc tất cả các lớp học từ tệp XML. Các phương thức để truy cập thông tin của từng lớp học, như lấy mã lớp và tên lớp.

Thêm, Xóa hoặc Sửa Dữ Liệu: Phương thức để thêm một lớp học mới vào tệp XML. Phương thức để xóa một lớp học cụ thể từ tệp XML và phương thức để cập nhật thông tin của một lớp học đã có.

3.3.3 Đối với tệp tin nganhhoc.xml

Đọc Dữ Liệu: Phương thức để đọc danh sách các ngành học từ tệp XML.

Thêm, Xóa hoặc Sửa Dữ Liệu: Có thể cung cấp phương thức để thêm, xóa, hoặc cập nhật thông tin về ngành học.

3.3.4 Đối với tệp tin users.xml

Đọc Dữ Liệu: Phương thức để đọc thông tin về người dùng từ tệp XML. Có thể cung cấp các phương thức để lọc thông tin theo id, mssv hoặc tên người dùng.

Thêm, Xóa hoặc Sửa Dữ Liệu: Phương thức để thêm một người dùng mới vào tệp XML.

Phương thức để xóa hoặc cập nhật thông tin của một người dùng đã có.

3.3.5 Mô hình khối DOM theo cấu trúc cây

Mô hình khối DOM (**Document Object Model**) trong XML được biểu diễn dưới dạng cây, trong đó mỗi phần tử, thuộc tính, và văn bản đều là các nút trong cây. Dưới đây là một mô hình khối DOM đơn giản cho một tài liệu XML lưu trữ thông tin của sinh viên:


```
<?xml version="1.0" encoding="UTF-8"?>

<students>

    <student mssv="121123132">

        <full_name>nguyen van a</full_name>

        <class_code>da21tta</class_code>

        <address>tra cu - tra vinh</address>

    </student>

</students>
```

Mô hình khối DOM:

```
Document (Tài liệu)
|
|-- students (Nhiều sinh viên)
|   |
|   |-- student (Một sinh viên)
|       |
|       |-- full_name (Tên sinh viên)
|           |
|           |-- Text: nguyen van a
|               |
|               |-- class_code (Mã lớp)
|                   |
|                   |-- Text: da21tta
|                       |
|                       |-- address (Địa chỉ)
|                           |
|                           |-- Text: tra cu - tra vinh
```

3.4 Thiết kế giao diện

3.4.1 Giao diện trang đăng nhập

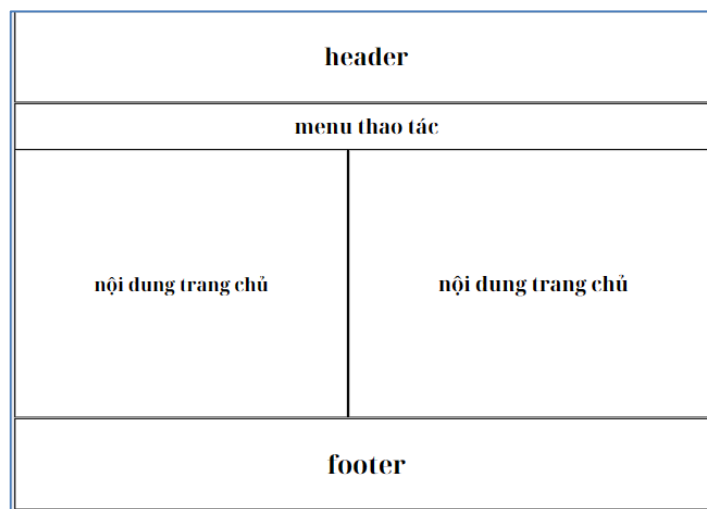


The diagram shows a login form titled "DĂNG NHẬP HỆ THỐNG". It contains two input fields: "Tên đăng nhập" (Username) and "Mật khẩu" (Password). Below the password field is a button labeled "Đăng nhập" (Login).

Hình 3.2 Thiết kế giao diện đăng nhập

Mô tả: Tôi thiết kế giao diện như trên, tạo một cảm giác đơn giản, dễ nhìn.

3.4.2 Giao diện trang quản lý thông tin sinh viên

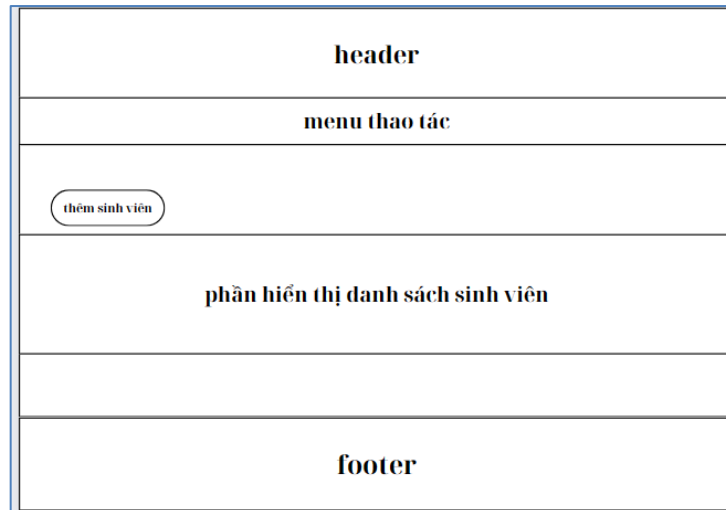


The diagram shows a main page layout with a header, a menu bar, and a footer. The menu bar is labeled "menu thao tác". The main content area is divided into two columns, both labeled "nội dung trang chủ".

Hình 3.3 Thiết kế giao diện chính

Mô tả: Giao diện chính bao gồm phần đầu trang hiển thị tên hệ thống, thanh menu gồm các trang chức năng thao tác với dữ liệu sinh viên. Phần lớn nhất chứa nội dung thông tin về Trường Đại Học Trà Vinh và cuối cùng phần chân trang hiển thị thông tin người thực hiện đồ án.

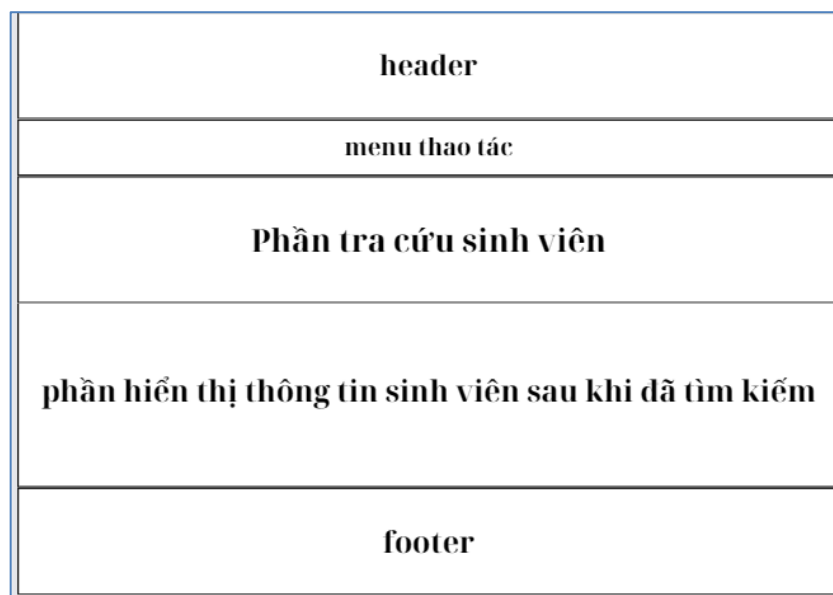
3.4.3 Giao diện xem danh sách sinh viên



Hình 3.4 Thiết kế giao diện xem danh sách sinh viên

Mô tả: Các phần đầu trang, chân trang và thanh menu vẫn giữ nguyên, thay đổi phần thân trang hiển thị danh sách sinh viên và có một nút thêm sinh viên mới vào danh sách sinh viên bên dưới.

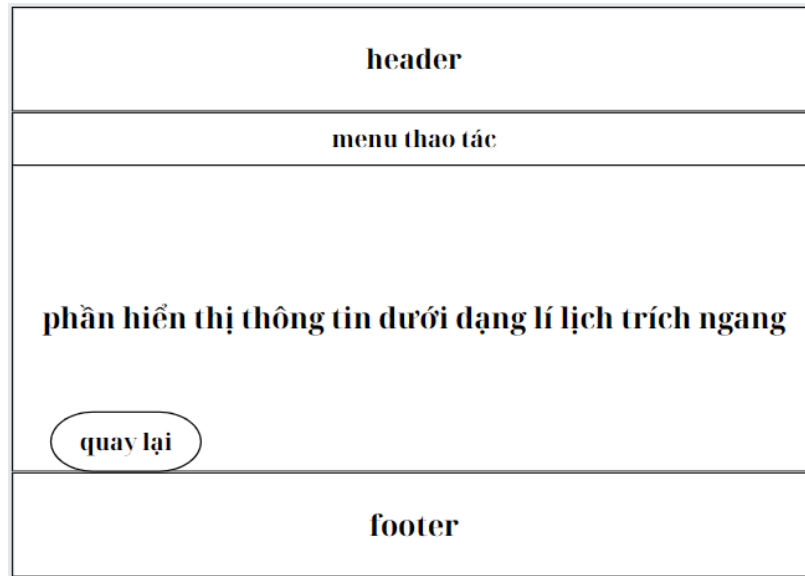
3.4.4 Giao diện trang tìm kiếm



Hình 3.5 Thiết kế giao diện tìm kiếm sinh viên

Mô tả: Ở phần này tôi chia phần thân trang ra làm 2 phần, một phần để nhập thông tin cần tra cứu và một phần để hiển thị danh sách sinh viên sau khi đã tra cứu.

3.4.5 Giao diện xem lý lịch sinh viên



Hình 3.6 *Thiết kế giao diện xem lý lịch trích ngang*

Mô tả: Phần xem lý lịch trích ngang của sinh viên này tôi lấy cả thân trang để hiển thị thông tin của sinh viên cần xem chi tiết và phía dưới thông tin sinh viên góc trái có một nút dùng để quay lại danh sách sinh viên.

CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU

4.1 Dữ liệu thử nghiệm

Danh sách các lớp học

Bảng 4.1 Bảng dữ liệu thử nghiệm lớp học

<i>MÃ LỚP</i>	<i>TÊN LỚP</i>
DA21TTA	CÔNG NGHỆ THÔNG TIN A
DA21TTB	CÔNG NGHỆ THÔNG TIN B
DA21TSA	NUÔI TRỒNG THỦY SẢN A
DA21TSB	NUÔI TRỒNG THỦY SẢN B
DA21CNOTA	CÔNG NGHỆ Ô TÔ A
DA21CNOTB	CÔNG NGHỆ Ô TÔ B
DA21TYA	THÚ Y A
DA21TYB	THÚ Y B
DA21RHMA	RĂNG HÀM MẶT A
DA21RHMB	RĂNG HÀM MẶT B

Danh sách các ngành học

Bảng 4.2 Bảng dữ liệu thử nghiệm ngành học

<i>STT</i>	<i>Tên ngành</i>
1	CÔNG NGHỆ THÔNG TIN
2	NUÔI TRỒNG THỦY SẢN
3	CÔNG NGHỆ Ô TÔ
4	THÚ Y
5	RĂNG HÀM MẶT
6	KẾ TOÁN
7	QUẢN TRỊ KINH DOANH
8	SƯ PHẠM NGŨ VĂN

Danh sách người dùng

Bảng 4.3 Bảng dữ liệu thử nghiệm tài khoản người dùng

<i>Id</i>	<i>Họ và tên</i>	<i>Tên tài khoản</i>	<i>Email</i>	<i>Quyền</i>
01	Cao duy nhân	caoduynhan	caoduynhan2003@gmail.com	1
02	Ngô thị trúc xuân	trucxuan	Trucxuan@gmail.com	1

Danh sách sinh viên

Bảng 4.4 Bảng dữ liệu thông tin sinh viên

<i>Mssv</i>	<i>Họ và tên</i>	<i>Mã lớp</i>	<i>Địa chỉ</i>	<i>Sdt</i>	<i>Email</i>	<i>Ngành học</i>	<i>Giới tính</i>	<i>Ngày sinh</i>	<i>Dân tộc</i>	<i>Tôn giáo</i>
110121246	Ngô Thị Trúc Xuân	da21tta	long đức – trà vinh	0359272876	Trucxuan@gmail.com	Công nghệ thông tin	Nu	12/2/1003	Kinh	Phật giáo
110121567	Cao Duy Nhân	Da21ttc	Trà cú – trà vinh	0329330518	caoduynhan2003@gmail.com	Công nghệ thông tin	Nam	23/6/2003	Kinh	Phật giáo

4.2 Kết quả thực nghiệm

4.2.1 Chức năng đăng nhập hệ thống

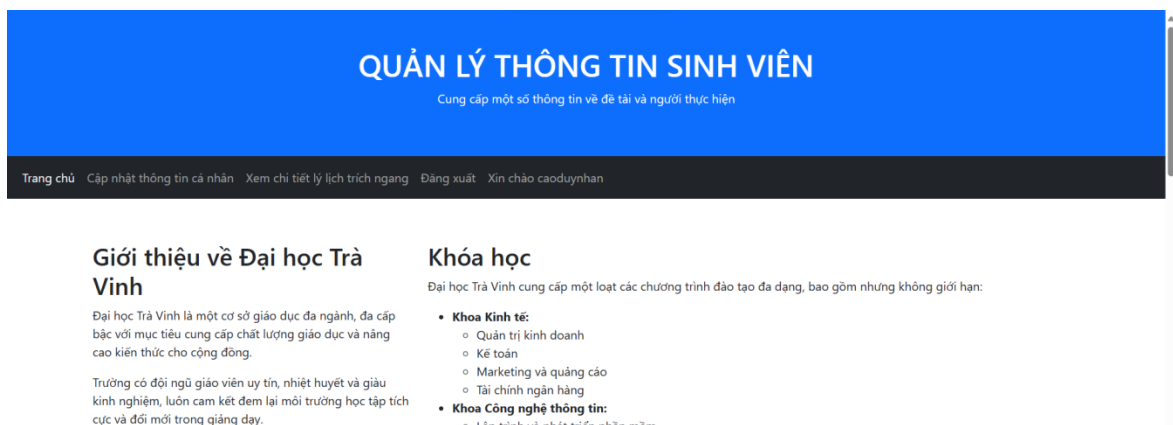
Đây là giao diện đăng nhập dành cho người dùng và quản trị:



Hình 4.1 Giao diện đăng nhập

Đây là phần bắt buộc trước khi vào hệ thống, cần có tài khoản chính xác của người dùng hoặc quản trị mà vào giao diện khác nhau để thực hiện các quyền có thể thao tác giữa quản trị và người dùng. Dựa vào mã số sinh viên (<mssv>) của mỗi người thì sẽ được cung cấp một tài khoản cá nhân để sinh viên có thể truy cập vào hệ thống để có thể chỉnh sửa thông tin của bản thân và xem thông tin đầy đủ dưới dạng lí lịch trích ngang của bản thân mình. Những tài khoản này sẽ được tổ chức trong file **users.xml** khi sinh viên đăng nhập sẽ duyệt theo mã số sinh viên của sinh viên đó để hiện đúng thông tin sinh viên đăng nhập.

Đây là giao diện trong trang dành cho người dùng:

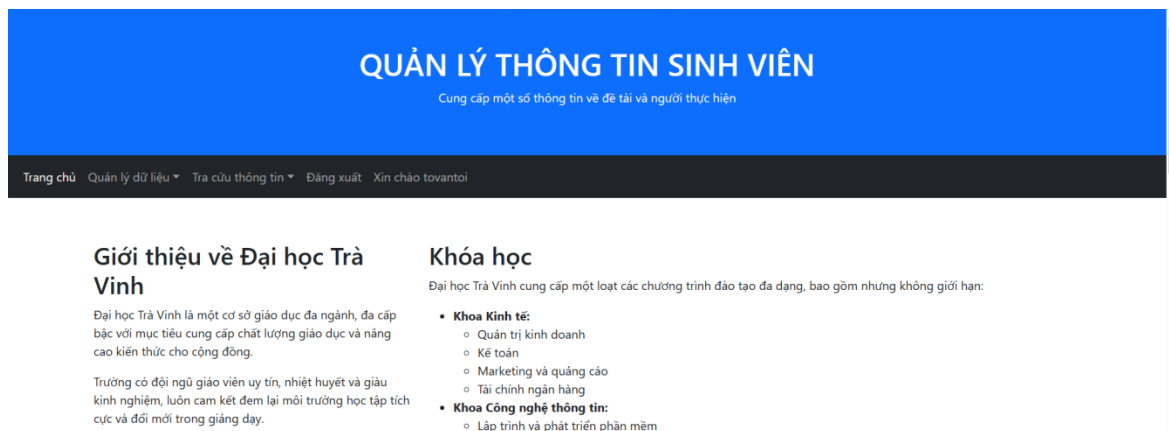


Hình 4.2 Giao diện người dùng

Ở trang giao diện người dùng có phần trang chủ nói sơ lược về Trường Đại Học Trà Vinh và có các chức năng dành cho người dùng như cập nhật lại thông tin

cá nhân hoặc chỉnh sửa lại khi bị sai sót và có phần xem thông tin đầy đủ của dưới dạng lí lịch trích ngang, cuối cùng là phần đăng xuất tài khoản người dùng.

Đây là giao diện dành cho quản trị:



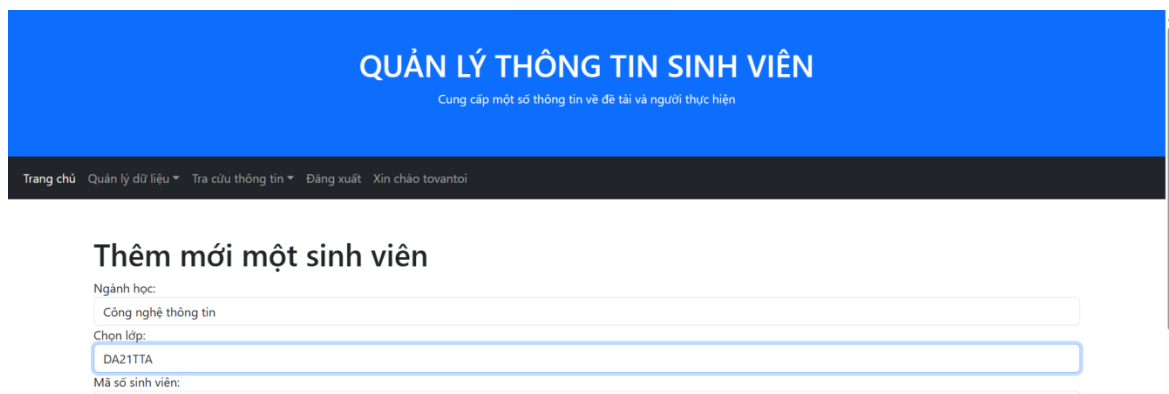
Hình 4.3 Giao diện quản trị

Giao diện dành cho quản trị cho phép người quản trị có các chức năng kiểm soát thông tin của sinh viên và có quyền thêm sửa xóa một số thông tin như: thông tin cá nhân, mã lớp, tên lớp, ngành học, tra cứu sinh viên theo lớp, tra cứu sinh viên theo ngành học và kiểm soát tài khoản người dùng.

4.2.2 Chức năng quản lý thông tin sinh viên

4.2.2.1 Thêm sinh viên

Người quản trị có khả năng thêm thông tin về sinh viên bằng cách nhập đầy đủ các trường dữ liệu mssv, họ tên, địa chỉ, ngành học.



Hình 4.4 Giao diện thêm sinh viên

Đây là giao diện thêm sinh viên của người quản trị. Người quản trị có khả năng thêm thông tin về sinh viên bằng cách nhập đầy đủ các trường dữ liệu mssv, họ tên, địa chỉ, ngành học,...Sau khi thêm dữ liệu sẽ được tạo và lưu vào file XML.

4.2.2.2 Sửa thông tin sinh viên

Người quản trị có khả năng chỉnh sửa thông tin sinh viên đã được lưu trữ, cho phép cập nhật các trường dữ liệu như địa chỉ, ngày sinh hoặc số điện thoại.

QUẢN LÝ THÔNG TIN SINH VIÊN
Cung cấp một số thông tin về đề tài và người thực hiện

[Trang chủ](#)
[Quản lý dữ liệu](#)
[Tra cứu thông tin](#)
[Đăng xuất](#)
[Xin chào tovanloi](#)

Chỉnh sửa Sinh viên

Ngành học:
Cổng nghệ thông tin

Chọn lớp:
Cổng nghệ thông tin C

Mã số sinh viên:
110121246

Tên sinh viên:
Ngô Thị Trúc Xuân


Địa chỉ:
duyên hải - trà vinh

Số điện thoại:
0359272229

Email:
trucxuan@gmail.com

Giới tính:
Nữ

Ngày sinh:
15/12/2023

Ảnh đại diện:


Dân tộc:
Kinh

Tôn Giáo:
phật giáo

Thông tin Phụ huynh

Họ và tên bố:
Phan Minh Thứ

Nghề nghiệp bố:
Kỹ sư phần mềm

Năm sinh bố:
1798

Họ và tên mẹ:
Kim Hoàng Nam

Nghề nghiệp mẹ:
Nội Trĩ

Năm sinh mẹ:
1999

Cập nhật

Quay lại

Thực tập Đồ án Cơ sở ngành
Lớp DA21TTA
Mã số sinh viên: 110121252 - Họ tên: Tô Văn Tới

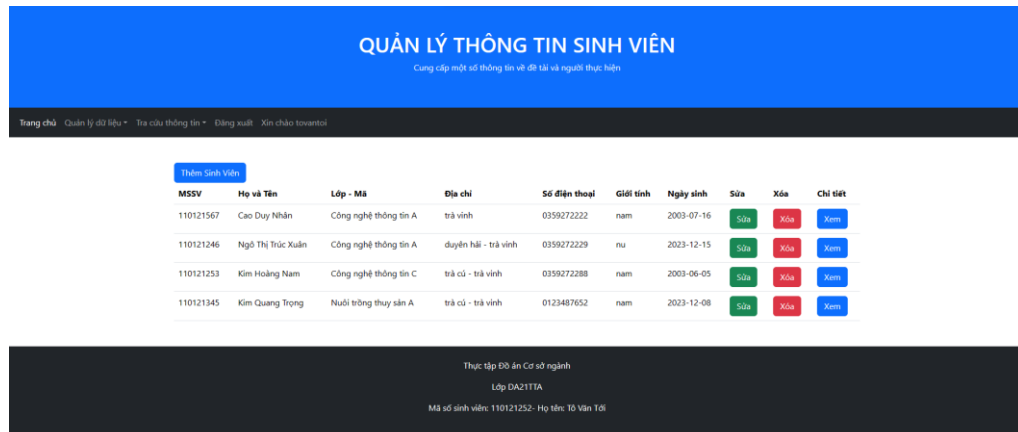
Hình 4.5 Giao diện sửa sinh viên

Đây là giao diện chỉnh sửa thông tin sinh viên của người quản trị, cho phép người quản trị có thể thay đổi thông tin trong các trường dữ liệu và cập nhật lại. Sau khi chỉnh sửa hoặc cập nhật dữ liệu sẽ được thay đổi và lưu lại trong file XML.

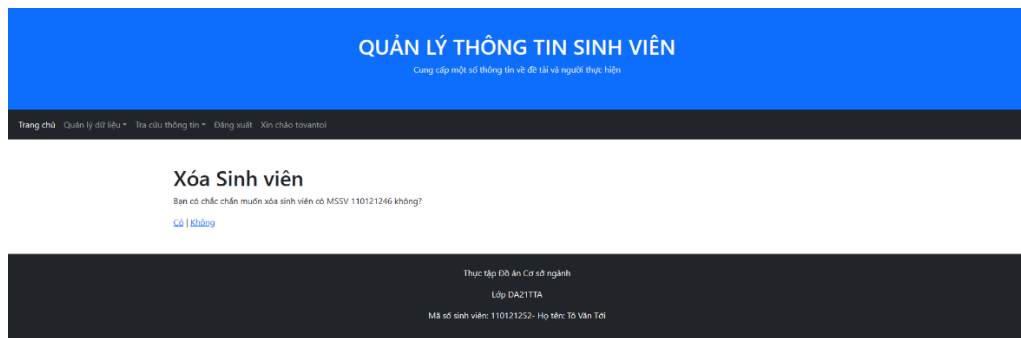
4.2.2.3 Xóa sinh viên

Người quản trị có thể xóa thông tin về sinh viên khỏi dữ liệu XML nếu sinh viên đây không còn được quản lý nữa.

Khi người dùng click vào nút xóa ở giao diện danh sách sinh viên, sau đó giao diện sẽ hiện ra một câu hỏi bạn có chắc muốn xóa sinh viên với mã số sinh viên muốn xóa không. Nếu người quản trị nhấn xóa thì sẽ tiến hành gọi đến hàm xóa, sau đó sẽ xóa dữ liệu người dùng với mã số sinh viên đã được chọn, nếu người quản trị nhấn không thì sẽ quay về giao diện danh sách sinh viên.

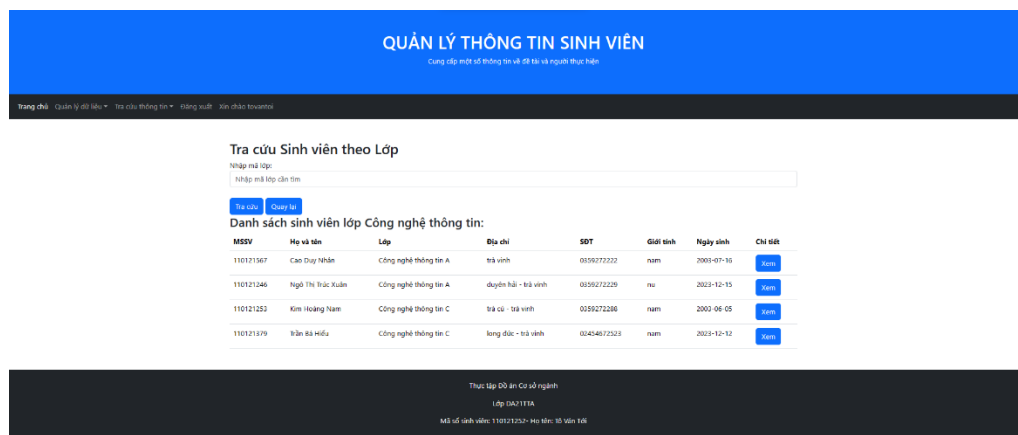


Hình 4.6 Giao diện xóa sinh viên



Hình 4.7 Giao diện xóa sinh viên

4.2.3 Tìm kiếm theo danh sách lớp



Hình 4.8 Giao diện tìm kiếm theo lớp

Đây là phần để người quản trị có thể tìm kiếm theo lớp một cách nhanh chóng, vì mỗi ngành học đều chia ra nhiều lớp khác nhau vì vậy đây là phần lọc ra những thông tin mà người quản trị cần tìm.

4.2.4 Tìm kiếm theo danh sách ngành học

Đây là phần tìm kiếm có quy mô rộng hơn danh sách lớp, tìm kiếm theo ngành học. Tất cả các sinh viên được tìm kiếm theo ngành học sẽ được hiện ra không phân biệt khác lớp như chức năng tìm kiếm theo lớp.

QUẢN LÝ THÔNG TIN SINH VIÊN
Cung cấp một số thông tin về dữ liệu và người thực hiện

Trang chủ Quản lý dữ liệu Tra cứu thông tin Đăng xuất Xin chào toantran

Tra cứu Sinh viên
Nhập tên ngành học:
Nhập mã lớp cần tìm

Danh sách sinh viên ngành Nuôi trồng thủy sản:

MSSV	Họ và tên	Lớp	Địa chỉ	SĐT	Giới tính	Ngày sinh	Chi tiết
110121345	Kim Quang Trung	Nuôi trồng thủy sản A	trà cổ - trà vinh	0123456789	nam	2023-12-08	Xem
110121543	Trần Nhật Hào	Nuôi trồng thủy sản B	trà vinh	0359272229	nam	2023-12-10	Xem
110121964	Nguyễn Thị Mỹ Tuyền	Nuôi trồng thủy sản A	duyên hải - trà vinh	0359272229	nu	2023-12-09	Xem

Thực tập Đồ Án Cơ sở ngành
Lớp DA21TTA
Mã số sinh viên: 110121252 - Họ tên: Tô Văn Tới

Hình 4.9 Giao diện tìm kiếm theo ngành

4.2.5 Cập nhật thông tin cá nhân của người dùng

Giống với chỉnh sửa và cập nhật của người quản trị, người dùng cũng có thể cập nhật thông tin của bản thân mình.

QUẢN LÝ THÔNG TIN SINH VIÊN
Cung cấp một số thông tin về dữ liệu và người thực hiện

Trang chủ Cập nhật thông tin cá nhân Xem chi tiết lý lịch trích ngang Đăng xuất Xin chào tranbahieu

Chỉnh sửa Sinh viên

Ngành học:
Công nghệ thông tin

Chọn lớp:
DA21TTA

Mã số sinh viên:
110121379

Tên sinh viên:
Trần Bá Hiếu

Địa chỉ:
long đức - trà vinh

Số điện thoại:
02454672523

Email:
hieuba@gmail.com

Giới tính:
Nam

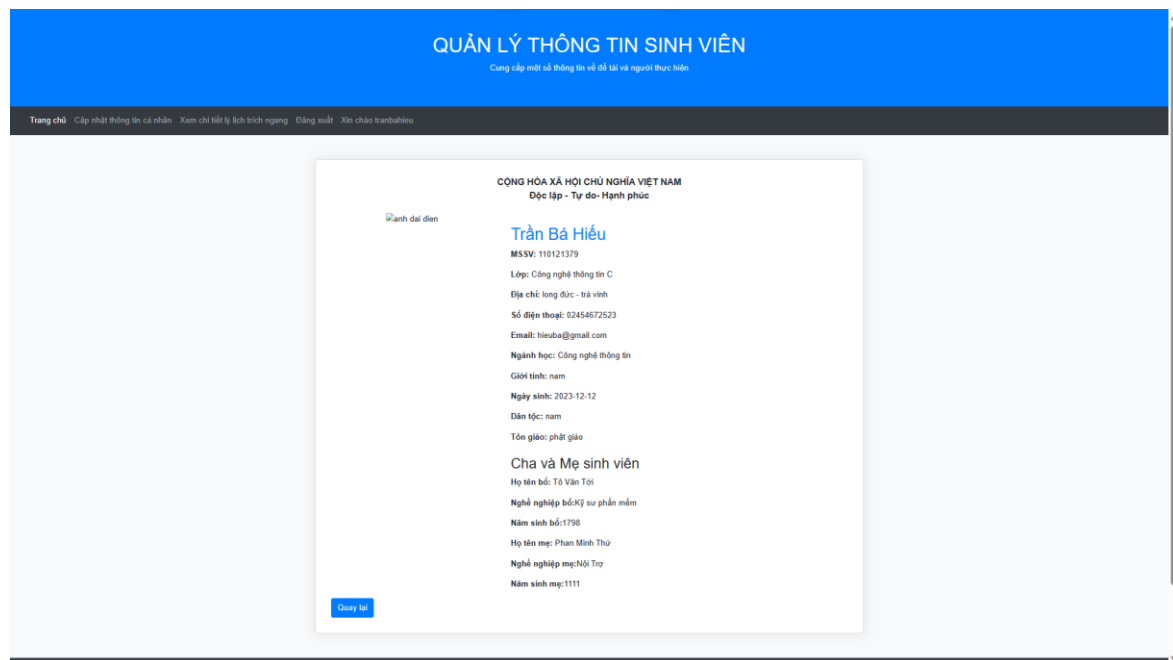
Ngày sinh:
12/12/2023

[Lưu](#)

Hình 4.10 Giao diện cập nhật thông tin

4.2.6 Xem lí lịch trích ngang của bản thân

Người dùng có thể xem thông tin đầy đủ của bản thân thông qua mục lí lịch trích ngang trên thanh menu.



Hình 4.11 Giao diện xem lí lịch trích ngang

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau thời gian làm đồ án, bản thân tôi tích lũy được một số kiến thức và kỹ năng sau:

- (1) Hiểu rõ cấu trúc và cách sử dụng ngôn ngữ XML;
- (2) Áp dụng XML để tổ chức lưu trữ thông tin sinh viên tiện lợi và hiệu quả;
- (3) Xây dựng hệ thống lưu trữ thông tin sinh viên dựa trên XML.

Ngoài ra, nghiên cứu này giúp tôi rèn luyện kỹ năng lập trình của bản thân, khả năng đọc hiểu và nghiên cứu tài liệu độc lập, khả năng quản lý thời gian.

5.2 Hướng phát triển

Cải thiện khả năng tương thích của giao diện website với đa dạng thiết bị di động. Mở rộng các tính năng nạp dữ liệu sinh viên từ nguồn dữ liệu bên ngoài, kết xuất các tập tin biểu mẫu liên quan đến thông tin sinh viên; Chức năng thống kê số lượng sinh viên theo ngành học, ...

Ngoài ra, hệ thống cần nâng cấp tính bảo mật dữ liệu, phương pháp mã hóa dữ liệu XML để có thể đảm bảo tính an toàn trong quá trình lưu trữ và truy cập dữ liệu một cách tốt nhất.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] N. P. Lan, XML nền tảng & ứng dụng, TP.HCM: LAO ĐỘNG XÃ HỘI, 2003.
- [2] T. N. T. Thúy, "PHP Là Gì? Giới Thiệu Về Ngôn Ngữ Lập Trình PHP Newbie Cần Biết," Glints, 16 06 2022. [Online]. Available: <https://glints.com/vn/blog/lap-trinh-php-la-gi/>. [Accessed 5 11 2023].
- [3] C. M. Nam, "Ngôn ngữ lập trình PHP," BLOG, 22 11 2021. [Online]. Available: <https://laptrinhtudau.com/php/>. [Accessed 17 11 2023].
- [4] L. Hoàng, "Bài 1: PHP là gì? Các khái niệm cơ bản cho người mới bắt đầu," STRINGEE, 3 8 2020. [Online]. Available: <https://stringee.com/vi/blog/post/PHP-la-gi-Cac-khai-niem-co-ban>. [Accessed 20 11 2023].