**ĐẠI HỌC PHENIKAA**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**⸎⸎⸎⸎⸎**



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN KĨ THUẬT PHẦN MỀM**

**Đề tài :** **HỆ THỐNG QUẢN LÝ SINH VIÊN**

Thành viên : Phạm Tùng Dương-23010113

Đoàn Đắc Tiến Đạt-23010789

Đặng Tuấn Mạnh-23010819

Võ Hoàng Dũng-23010828

Trịnh Thị Thảo -23010804

Giáo viên hướng dẫn : TS.Vũ Quang Dũng

Lớp tín chỉ : Kĩ thuật phần mềm-1-2-23 (COUR01.LT2)

**HÀ NỘI, NGÀY 20 THÁNG 6 NĂM 2025**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

..............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

|  |  |
| --- | --- |
|  | ............., ngày... tháng... năm 2025  Xác nhận của giảng viên hướng dẫn  *(Ký và ghi rõ họ tên)* |

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 4](#_Toc201252545)

[DANH MỤC BẢNG 5](#_Toc201252546)

[LỜI MỞ ĐẦU 7](#_Toc201252547)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 9](#_Toc201252548)

[1.1. Tìm hiểu đề tài 9](#_Toc201252549)

[1.2. Mục tiêu tài liệu 9](#_Toc201252550)

[1.3. Tổng quan hệ thống 10](#_Toc201252551)

[1.4. Cơ sở lý thuyết 11](#_Toc201252552)

[CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 22](#_Toc201252553)

[2.1 Phân tích hệ thống 22](#_Toc201252554)

[2.1.1. Sơ đồ phân rã chức năng 22](#_Toc201252555)

[2.1.2. Sơ đồ luồng dữ liệu ngữ cảnh 23](#_Toc201252556)

[2.1.3 Sơ đồ luồng dữ liệu mức đỉnh 23](#_Toc201252557)

[2.1.4 Sơ đồ mức dưới đỉnh của Quản lý tài khoản 25](#_Toc201252558)

[2.1.5. Sơ đồ mức dưới đỉnh của Quản lý sản phẩm 25](#_Toc201252559)

[2.1.6. Sơ đồ mức dưới đỉnh của Quản lý hóa đơn và thanh toán 26](#_Toc201252560)

[2.1.7. Sơ đồ mức dưới đỉnh của Quản lý phản hồi 27](#_Toc201252561)

[2.2. Biểu đồ cơ sở dữ liệu 28](#_Toc201252562)

[2.3 Mô tả bảng cơ sở dữ liệu 28](#_Toc201252563)

[2.4. Biểu đồ use case tổng quát 32](#_Toc201252564)

[2.5. Đặc tả use case 32](#_Toc201252565)

[2.5.1. Use case đăng nhập 32](#_Toc201252566)

[2.5.2. Use case đăng ký 37](#_Toc201252567)

[2.5.3. Use case tìm kiếm 41](#_Toc201252568)

[CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG 43](#_Toc201252569)

[3.1. Quá trình triển khai hệ thống 43](#_Toc201252570)

[3.2. Giao diện một số hệ thống 45](#_Toc201252571)

[KẾT LUẬN 48](#_Toc201252572)

[TÀI LIỆU THAM KHẢO 49](#_Toc201252573)

# DANH MỤC HÌNH ẢNH

[Hình 1 Logo Javascript 15](#_Toc201262242)

[Hình 2 Sơ đồ phân rã chức năng 22](#_Toc201262243)

[Hình 3 Sơ đồ luồng dữ liệu ngữ cảnh 23](#_Toc201262244)

[Hình 4 Sơ đồ luồng dữ liệu mức đỉnh 24](#_Toc201262245)

[Hình 5 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý tài khoản 25](#_Toc201262246)

[Hình 6 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý sản phẩm 26](#_Toc201262247)

[Hình 7 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý hóa đơn và thanh toán 27](#_Toc201262248)

[Hình 8 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý phản hồi 27](#_Toc201262249)

[Hình 10 Biểu đồ cơ sở dữ liệu 28](#_Toc201262250)

[Hình 11 Biểu đồ use case tổng quát 32](#_Toc201262251)

[Hình 12 Biểu đồ use case đăng nhập 33](#_Toc201262252)

[Hình 13 Biểu đồ trình tự đăng nhập 35](#_Toc201262253)

[Hình 14 Biểu đồ hoạt động đăng nhập 36](#_Toc201262254)

[Hình 15 Biểu đồ use case đăng ký 37](#_Toc201262255)

[Hình 16 Biểu đồ trình tự đăng ký 39](#_Toc201262256)

[Hình 17 Biểu đồ hoạt động đăng ký 40](#_Toc201262257)

[Hình 18 Biểu đồ use case tìm kiếm 41](#_Toc201262258)

[Hình 28 Giao diện trang chủ 45](#_Toc201262259)

[Hình 29 Giao diện trang đăng nhập 46](#_Toc201262260)

[Hình 30 Giao diện trang đăng kí 47](#_Toc201262261)

[Hình 31 Giao diện chức năng tìm kiếm, đăng ký môn học 47](#_Toc201262262)

# DANH MỤC BẢNG

[Bảng 1.Bảng user 31](#_Toc200461539)

[Bảng 2.Bảng wishlist 31](#_Toc200461540)

[Bảng 3.Bảng products 32](#_Toc200461541)

[Bảng 4.Bảng oders 32](#_Toc200461542)

[Bảng 5.Bảng message 33](#_Toc200461543)

[Bảng 6.Bảng cart 34](#_Toc200461544)

DANH MỤC KÝ TỰ VÀ CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Viết tắt** | **Diễn giải** |
| PHP | Hypertext Preprocessor – Ngôn ngữ lập trình phía máy chủ |
| HTML | HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản |
| CSS | Cascading Style Sheets – Ngôn ngữ định kiểu cho trang web |
| JS | JavaScript – Ngôn ngữ lập trình phía máy khách, dùng để tạo tương tác động |
| DB | Database – Cơ sở dữ liệu |
| UI | User Interface – Giao diện người dùng |
| UX | User Experience – Trải nghiệm người dùng |
| SQL | Structured Query Language – Ngôn ngữ truy vấn dữ liệu |
| CRUD | Create, Read, Update, Delete – Các thao tác cơ bản với dữ liệu |
| MVC | Model - View - Controller – Mô hình kiến trúc phần mềm |
| XAMPP | Cross-Platform, Apache, MySQL, PHP, Perl – Gói phần mềm máy chủ cục bộ |
| AJAX | Asynchronous JavaScript and XML – Kỹ thuật truyền dữ liệu bất đồng bộ |
| CDN | Content Delivery Network – Mạng phân phối nội dung |
| IDE | Integrated Development Environment – Môi trường phát triển tích hợp |
| MVC | Model - View - Controller – Kiến trúc điều khiển trong ứng dụng web |
| UX/UI | User Experience / User Interface – Trải nghiệm và giao diện người dùng |
| SSL | Secure Sockets Layer – Giao thức bảo mật truyền thông |

# LỜI MỞ ĐẦU

**Trong kỷ nguyên công nghệ số phát triển mạnh mẽ hiện nay, việc ứng dụng các giải pháp phần mềm vào công tác quản lý đã và đang trở thành xu hướng tất yếu trong mọi lĩnh vực, đặc biệt là lĩnh vực giáo dục. Ở các trường đại học, cao đẳng, việc quản lý thông tin sinh viên, lớp học, môn học, điểm số... nếu thực hiện thủ công hoặc chỉ dựa vào các bảng tính rời rạc sẽ không tránh khỏi những bất cập như mất thời gian, dễ xảy ra sai sót và khó khăn trong việc tra cứu, tổng hợp dữ liệu.**

**Xuất phát từ thực tiễn đó, đề tài “Xây dựng hệ thống quản lý sinh viên” không chỉ là một nhiệm vụ trong khuôn khổ học phần Kỹ thuật phần mềm, mà còn là cơ hội để nhóm sinh viên tiếp cận quy trình phát triển phần mềm một cách bài bản – từ khâu phân tích yêu cầu, thiết kế giao diện, xây dựng tính năng cho đến kiểm thử và triển khai thực tiễn.**

**Với mong muốn xây dựng một nền tảng quản lý hiện đại, thân thiện với người dùng và đảm bảo tính chính xác, bảo mật thông tin, nhóm em lựa chọn phát triển hệ thống quản lý sinh viên nhằm góp phần nâng cao hiệu quả công tác quản lý, tiết kiệm nguồn lực cho nhà trường, đồng thời mang lại trải nghiệm tiện lợi cho giáo viên và sinh viên trong việc tra cứu, cập nhật thông tin học tập.**

**Lí do chọn đề tài:**

Trong bối cảnh chuyển đổi số mạnh mẽ hiện nay, việc quản lý thông tin sinh viên tại các trường học vẫn còn nhiều bất cập nếu thực hiện thủ công hoặc chỉ sử dụng bảng tính Excel. Công tác lưu trữ, tìm kiếm, cập nhật và thống kê dữ liệu thường gặp khó khăn, dễ xảy ra sai sót hoặc mất mát thông tin. Điều này gây ảnh hưởng đến hiệu quả quản lý và chất lượng phục vụ sinh viên.

Đề tài "*Hệ thống Quản lý Sinh viên*" được lựa chọn với mục tiêu:

- Tin học hóa toàn bộ quá trình quản lý sinh viên, lớp học, môn học, điểm số… tại nhà trường.

- Giúp cán bộ quản lý, giáo viên và sinh viên dễ dàng tra cứu, cập nhật, tổng hợp và xuất báo cáo một cách nhanh chóng, chính xác.

- Nâng cao hiệu quả quản lý, tiết kiệm thời gian, giảm chi phí và hạn chế tối đa các sai sót trong nghiệp vụ.

- Đáp ứng yêu cầu thực tế của các trường học hiện đại, tạo tiền đề mở rộng các chức năng mới trong tương lai (quản lý học phí, hạnh kiểm, thông báo…).

Với các lý do trên, nhóm quyết định chọn đề tài "Hệ thống Quản lý Sinh viên" làm bài tập lớn nhằm rèn luyện kỹ năng phân tích, thiết kế, triển khai ứng dụng quản lý thực tế.

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Tìm hiểu đề tài

Trong những năm gần đây, chuyển đổi số trong lĩnh vực giáo dục tại Việt Nam đã diễn ra mạnh mẽ, tạo ra nhiều cơ hội để các trường học và tổ chức giáo dục nâng cao hiệu quả quản lý, giảng dạy cũng như phục vụ sinh viên. Việc quản lý sinh viên – bao gồm thông tin cá nhân, lớp học, môn học, điểm số và các hoạt động liên quan – đóng vai trò quan trọng trong hoạt động của mỗi nhà trường. Tuy nhiên, nếu thực hiện thủ công hoặc bằng các bảng tính rời rạc, quá trình này thường gặp nhiều bất cập như sai sót, khó kiểm soát, mất nhiều thời gian và công sức.

Việc xây dựng một hệ thống quản lý sinh viên hiện đại giúp nhà trường dễ dàng lưu trữ, tìm kiếm, cập nhật và thống kê thông tin một cách trực quan, nhanh chóng, đồng thời hỗ trợ cán bộ quản lý, giáo viên và sinh viên tương tác hiệu quả hơn. Người dùng có thể dễ dàng tra cứu thông tin, xem kết quả học tập, đăng ký môn học, gửi phản hồi hoặc cập nhật hồ sơ cá nhân chỉ với vài thao tác đơn giản. Bên cạnh đó, hệ thống còn hỗ trợ nhà trường trong việc quản lý lớp học, môn học, phân công giảng dạy, theo dõi tiến độ và xuất báo cáo tổng hợp dễ dàng.

Đề tài “Xây dựng hệ thống quản lý sinh viên” không chỉ có ý nghĩa thực tiễn cao mà còn giúp sinh viên tiếp cận toàn diện quy trình phát triển một hệ thống phần mềm từ khâu phân tích yêu cầu, thiết kế, lập trình đến triển khai và kiểm thử. Đây là mô hình phù hợp với xu thế hiện đại, đồng thời góp phần rèn luyện kỹ năng xây dựng sản phẩm ứng dụng thực tế cho nhóm sinh viên.

## Mục tiêu tài liệu

- Cung cấp hướng dẫn chi tiết cho quá trình xây dựng một hệ thống quản lý sinh viên đơn giản, từ giai đoạn phân tích, thiết kế, phát triển cho đến triển khai thực tế. Báo cáo giúp người đọc hình dung toàn bộ quy trình phát triển phần mềm quản lý sinh viên theo phương pháp bài bản.

- Giúp người đọc nắm bắt rõ hơn về cấu trúc hệ thống, cơ sở dữ liệu, giao diện người dùng, cũng như các công nghệ chính được sử dụng để xây dựng hệ thống quản lý sinh viên, bao gồm cách tổ chức các chức năng quản lý thông tin sinh viên, lớp học, môn học, điểm số, phân quyền truy cập...

- Ghi nhận những kinh nghiệm thực tế trong quá trình phát triển, bao gồm cả những khó khăn gặp phải (ví dụ: thiết kế giao diện thân thiện, tối ưu truy vấn cơ sở dữ liệu, xử lý phân quyền) và các giải pháp kỹ thuật đã được áp dụng để xử lý vấn đề.

- Là tài liệu tham khảo hữu ích cho các sinh viên, học viên hoặc cá nhân đang tìm hiểu về thiết kế, xây dựng hệ thống quản lý sinh viên bằng các ngôn ngữ lập trình PHP, HTML/CSS, JavaScript, Bootstrap, cũng như cách tổ chức dự án và triển khai ứng dụng thực tế.

## 1.3. Tổng quan hệ thống

Hệ thống quản lý sinh viên được xây dựng với mục tiêu phục vụ công tác quản lý, tra cứu và hỗ trợ học tập cho sinh viên, giáo viên và cán bộ nhà trường một cách hiện đại, hiệu quả và tiện lợi. Hệ thống cung cấp các chức năng cơ bản nhưng đầy đủ để hỗ trợ cả người dùng lẫn người quản trị như sau:

- Giao diện trang chủ thân thiện, hiển thị các thông báo quan trọng, lịch học, sự kiện nổi bật và các tin tức liên quan đến sinh viên.

- Chức năng tìm kiếm và lọc thông tin sinh viên theo tên, mã số, lớp, khoa, khoá học hoặc trạng thái học tập.

- Quản lý hồ sơ cá nhân: Sinh viên có thể đăng ký tài khoản, đăng nhập và quản lý thông tin cá nhân, theo dõi kết quả học tập, đăng ký môn học, xem lịch học và lịch thi.

- Quản lý học tập: Giáo viên và cán bộ có thể nhập điểm, cập nhật kết quả học tập, xét học bổng, cảnh báo học vụ, và xuất các báo cáo liên quan.

- Quản lý lớp học, môn học: Người quản trị có thể thêm, sửa, xóa thông tin lớp học, môn học, phân công giảng viên, quản lý danh sách sinh viên theo lớp hoặc môn.

- Xử lý thông báo, phản hồi: Hệ thống hỗ trợ gửi thông báo đến sinh viên, nhận phản hồi hoặc ý kiến đóng góp từ sinh viên, phụ huynh, giáo viên.

- Thống kê, báo cáo: Cung cấp các báo cáo thống kê về số lượng sinh viên, kết quả học tập, tỷ lệ tốt nghiệp, điểm trung bình của từng lớp, môn hoặc khoa.

- Tương thích tốt với các thiết bị di động, máy tính bảng và trình duyệt phổ biến, tối ưu giao diện bằng Bootstrap.

- Tích hợp chatbot hỗ trợ sinh viên trong việc tra cứu thông tin, hướng dẫn sử dụng các chức năng hệ thống, trả lời các câu hỏi thường gặp về quy chế, thủ tục.

Hệ thống được xây dựng nhằm mục tiêu đem đến trải nghiệm quản lý học tập trực tuyến hiện đại, an toàn và thân thiện, đặc biệt tập trung vào việc hỗ trợ sinh viên, giáo viên và cán bộ quản lý trong môi trường giáo dục số hóa ngày nay.

## 1.4. Cơ sở lý thuyết

1.4. 1 Ngôn ngữ UML

UML là viết tắt của “Unified Modeling Language” - là ngôn ngữ mô hình hóa thống nhất, dùng để đặc tả, hình dung, xây dựng và tư liệu hóa phần mềm hướng đối tượng.

UML dùng để:

* Trực quan hoá: Đưa ra các mô hình trực quan về hệ thống để mọi người có thể hiểu được.
* Đặc tả: UML cho phép xây dựng các mô hình đúng đắn và đầy đủ để mô tả hệ thống.
* Xây dựng: Các mô hình UML có thể liên kết trực tiếp với nhiều ngôn ngữ lập trình Java, C++, VB, ...
* Lập tài liệu: UML cho phép tạo các tài liệu như tài liệu về các yêu cầu của người dùng, kiến trúc hệ thống, thiết kế hệ thống, kiểm thử, kế hoạch dự án, ...

UML có thể được sử dụng làm công cụ giao tiếp giữa người dùng, nhà phân tích, nhà thiết kế và nhà phát triển phần mềm.

Các phần tử của UML:

View (Quan sát): Theo các phương diện khác nhau của hệ thống cần phân tích, thiết kế. Dựa vào các quan sát để thiết lập kiến trúc cho hệ thống cần phát triển.

***Có 9 loại biểu đồ:***

* Use Case Diagram (Biểu đồ ca sử dụng): Mô tả sự tương tác giữa các tác nhân và hệ thống thông qua các ca sử dụng.
* Class Diagram (Biểu đồ lớp): Biểu đồ lớp chỉ ra sự tồn tại giữa các lớp và mối quan hệ giữa chúng trong thiết kế logic trong hệ thống. Khung nhìn tĩnh của hệ thống chủ yếu hỗ trợ các chức năng của hệ thống.
* Object Diagram (Biểu đồ đối tượng): Biểu đồ đối tượng đại diện cho một thể hiện của biểu đồ lớp, cho nên các khái niệm cơ bản là giống biểu đồ lớp. Biểu đồ đối tượng được sử dụng để hiện thị một tập hợp các đối tượng và các mối quan hệ của chúng như một thể hiện.
* Sequence Diagram (Biểu đồ trình tự): Là biểu đồ để xác định và chỉ rõ vai trò của đối tượng tham gia vào luồng sự kiện của use case.
* Collaboration Diagram (Biểu đồ cộng tác): Tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
* State Diagram (Biểu đồ trạng thái): Thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống.
* Activity Diagram (Biểu đồ hành động): Được sử dụng để mô tả các hoạt động và các hành động được thực hiện trong một use case.
* Component Diagram (Biểu đồ thành phần): Chỉ ra cấu trúc vật lý của các thành phần trong hệ thống.
* Deployment Diagram (Biểu đồ triển khai): Chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

Relationship (Quan hệ):

* Realization (Hiện thực hoá):Là quan hệ ngữ nghĩa giữa giao diện và lớp.
* Dependency (Phụ thuộc):Là quan hệ ngữ nghĩa giữa hai phần tử, trong đó sự thay đổi của một tử sẽ tác động đến ngữ nghĩa của phần tử phụ thuộc.
* Generalization (Kế thừa): Là quan hệ mô tả sự khái quát hoá mà trong đó một số đối tượng cụ thể (của lớp con) sẽ được kế thừa các thuộc tính, các phương thức của các đối tượng tổng quát (lớp cơ sở).
* Association (Kết hợp): Là quan hệ cấu trúc xác định mối liên kết giữa các lớp đối tượng.

1.4. 2 Ngôn ngữ HTML

* HTML là gì?

HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo cấu trúc và hiển thị nội dung trên trang web. Tuy nhiên, không có một câu chuyện cụ thể nào về việc đặt tên cho HTML giống như việc đặt tên cho ngôn ngữ lập trình Java.

HTML được tạo ra bởi Sir Tim Berners-Lee, một nhà khoa học máy tính người Anh, vào năm 1990. Tên "HTML" được đặt ra để mô tả chức năng cơ bản của ngôn ngữ, đó là tạo ra các trang web dưới dạng siêu văn bản (hypertext), nơi mà các trang web có thể kết nối với nhau thông qua các liên kết (hyperlinks).

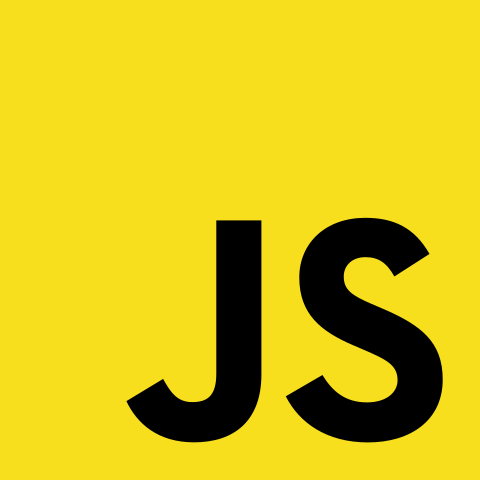
Trong HTML, các phần tử được sử dụng để đánh dấu và định nghĩa cấu trúc của một trang web, bao gồm văn bản, hình ảnh, liên kết và các phần tử tương tác khác. Dù không có một câu chuyện cụ thể về việc đặt tên cho HTML giống như Java, tên "HTML" đã trở thành biểu tượng của trang web và là một phần quan trọng của ngôn ngữ lập trình web hiện đại.

* Ứng dụng của HTML

HTML (Hypertext Markup Language) thường được sử dụng để tạo nội dung và cấu trúc trang web, trong khi các ngôn ngữ lập trình như Java được sử dụng để phát triển các ứng dụng và chức năng phức tạp hơn. Tuy nhiên, HTML có thể được sử dụng kết hợp với các ngôn ngữ lập trình khác để tạo ra các ứng dụng web đa dạng và phong phú.

Một số ví dụ về cách HTML có thể được sử dụng trong các ứng dụng khác nhau:

* Trang web tĩnh: HTML được sử dụng để tạo các trang web tĩnh với nội dung cố định và không thay đổi theo thời gian. Điều này bao gồm các trang web thông tin, trang giới thiệu sản phẩm, và các trang cá nhân.
* Web Applets: HTML có thể được sử dụng để nhúng các ứng dụng Java (applets) vào trong trang web. Điều này cho phép tạo ra các trang web có chức năng động hơn, bao gồm các trò chơi trực tuyến, biểu đồ tương tác, và các ứng dụng web phức tạp.
* Form và Giao diện người dùng: HTML được sử dụng để tạo ra các biểu mẫu và giao diện người dùng cho các trang web. Các biểu mẫu HTML có thể chứa các trường đầu vào, nút gửi và các phần tử khác để thu thập thông tin từ người dùng.
* Email Templates: HTML cũng được sử dụng để tạo các mẫu email, cho phép tạo ra các email có định dạng, hình ảnh và liên kết phong phú hơn.
* Web Scraping và Parsing: HTML có thể được sử dụng để phân tích và trích xuất dữ liệu từ các trang web khác, thông qua kỹ thuật web scraping và parsing. Điều này cho phép tự động hóa việc thu thập thông tin từ internet cho mục đích phân tích dữ liệu hoặc xây dựng ứng dụng.
* Tích hợp với CSS và JavaScript: HTML thường được kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra các trang web đa dạng, đẹp mắt và chức năng. CSS được sử dụng để thiết kế và trang trí các phần tử HTML, trong khi JavaScript được sử dụng để thêm chức năng tương tác và động cho các trang web.

1.4. 3 Ngôn ngữ JavaScript

Hình 1 Logo Javascript

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). JavaScript chứa các thư viện tiêu chuẩn cho các đối tượng (object), ví dụ như: Array, Date, Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

* Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* Server-side JavaScript - JavaScript phía máy chủ, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

JavaScript được phát triển bởi Brendan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được biết đến lần đầu tiên là Mocha, rồi sau đó được đổi tên thành LiveScript và cuối cùng là JavaScript nổi tiếng như hiện giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian.

Năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lúc đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web.

JavaScript cùng với HTML, CSS đã trở thành ngôn ngữ không thể thiếu trong lập trình web hiện nay:

* Với việc phát triển mạnh mẽ của công nghệ, JavaScript đã dần trở thành một ngôn ngữ đa nền tảng với nhiều ứng dụng khác nhau.
* Các Framework, Library cho lập trình front-end mạnh mẽ hỗ trợ xây dựng các trang Single-page application (SPA) như: ReactJS, Angular, VueJS.
* Hỗ trợ lập trình mobile bằng thư viện React Native.
* Hỗ trợ lập trình Window bằng Electron.
* Không chỉ phát triển ở khía cạnh lập trình phía client-side, JavaScript cũng hỗ trợ viết server-side rất mạnh mẽ với NodeJS.

Với hệ sinh thái mã nguồn mở khổng lồ cùng sự phát triển như hiện nay, JavaScript được dự đoán sẽ phát triển mạnh mẽ hơn nữa trong những năm tới.

1.4. 4 Ngôn ngữ PHP

* + - * PHP là gì?
* PHP (Hypertext Preprocessor) là một ngôn ngữ kịch bản phía server được thiết kế đặc biệt để phát triển các ứng dụng web. PHP là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực lập trình web, cho phép các lập trình viên tạo ra các trang web động một cách dễ dàng và hiệu quả.
  + - * Lịch sử phát triển của PHP
* PHP được phát triển bởi Rasmus Lerdorf vào năm 1994. Ban đầu, nó chỉ là một tập hợp các tập lệnh CGI viết bằng ngôn ngữ C nhằm theo dõi người dùng truy cập website cá nhân của Rasmus. Tuy nhiên, sau đó nó đã phát triển thành một ngôn ngữ kịch bản mạnh mẽ, cho phép tạo ra các ứng dụng web động và tương tác.
* Kể từ khi ra đời, PHP đã trải qua nhiều phiên bản cải tiến với các tính năng và công cụ hỗ trợ phát triển website. Hiện nay, PHP vẫn là ngôn ngữ hàng đầu được sử dụng trong nhiều nền tảng mã nguồn mở như WordPress, Drupal, Joomla, và nhiều hệ thống quản lý nội dung khác.
  + - * Mục tiêu của PHP
* Mục tiêu chính của PHP là cung cấp một ngôn ngữ đơn giản, linh hoạt và hiệu quả để phát triển các ứng dụng web động. Các mục tiêu cụ thể của PHP bao gồm:
* Xử lý phía server: PHP được thiết kế để xử lý logic phía server, làm việc với cơ sở dữ liệu và tạo ra các trang web động.
* Tích hợp dễ dàng với HTML: PHP được nhúng trực tiếp vào mã HTML, giúp lập trình viên dễ dàng xây dựng các ứng dụng web phức tạp mà không cần phải học nhiều ngôn ngữ khác.
* Tương thích cao: PHP tương thích với hầu hết các hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL, SQLite, giúp dễ dàng lưu trữ và quản lý dữ liệu của ứng dụng web.
* Đơn giản và hiệu quả: PHP được thiết kế để dễ học và sử dụng, giúp các lập trình viên nhanh chóng tạo ra các ứng dụng web mà không cần phải cấu hình phức tạp.
* Tương tác với người dùng: PHP cho phép nhận dữ liệu từ người dùng thông qua các form nhập liệu HTML, sau đó xử lý và trả về kết quả tương ứng.

1.4. 5 Cơ sở dữ liệu

XAMPP là gì?

Phần mềm Xampp là một trong những phần mềm được nhiều lập trình viên sử dụng để thiết lập website theo ngôn ngữ PHP. XAMPP có công dụng thiết lập web server có cài đặt sẵn các công cụ như PHP, Apache, MySQL… Xampp sở hữu thiết kế giao diện thân thiện với người dùng, cho phép các lập trình viên có thể đóng mở hoặc reboot các tính năng của server mọi lúc. Ngoài ra, Xampp cũng được xây dựng theo source code mở.

Đánh giá một cách tổng thể, Xampp được đánh giá là một trong những phần mềm trọng yếu trong việc xây dựng website dành cho các lập trình viên PHP. Xampp cài đặt các thành tố trọng yếu, hỗ trợ lẫn nhau bao gồm:

* Apache
* PHP (thiết lập nền tảng để các tập tin script \*.php hoạt động);
* MySql (hệ quản trị dữ liệu)
* Perl

Các thành chính của XAMPP

* Apache:

Apache là một website software sử dụng source code mở hoàn toàn miễn phí, được sử dụng bởi 46% trên tổng số các website trên toàn cầu. Phần mềm web này được thiết lập và cập nhật bởi Apache Software Foundation.

Ra đời hơn 20 năm về trước, Apache được nhiều chuyên gia công nhận là một trong những web server uy tín, đáng sử dụng cho các website. Song song với NGINX (một phần mềm web phổ biến khác), Apache giúp website sở hữu một server hoàn thiện hơn, có thể tải nhiều nội dung lên cho website của mình nổi bật hơn mà không gặp phải bất kỳ trở ngại nào.

* Mysql:

MySQL là một hệ thống quản lý cơ sở dữ liệu liên kết có ưu điểm nhanh chóng, dễ dùng cho các lập trình viên. MySQL được ứng dụng khá nhiều cho các website thương mại điện tử hiện nay.

Hệ thống được phát triển bởi công ty Thụy Điển này hiện được nhiều lập trình viên hàng đầu trên thế giới ưa chuộng bởi những ưu điểm sau.

MySQL là một opensource, hoàn toàn miễn phí.

Hiệu suất hoạt động của MySQL rất mạnh mẽ, kể cả khi khối lượng dữ liệu khổng lồ. Có thể kiểm tra toàn vẹn dữ liệu với SQL Trigger.

Cái tên nói lên tất cả, hệ thống này sử dụng ngôn ngữ dữ liệu SQL để cấu trúc hóa website của bạn.

MySQL tương thích với hầu hết các hệ điều hành (Windows, Mac, Linux…) cùng các ngôn ngữ lập trình như PHP, C++, Java…

MySQL rất tương thích với PHP, ngôn ngữ lập trình phổ biến của nhiều lập trình viên.

MySQL có hệ thống thông tin khổng lồ, chứa đến 50 triệu hàng dữ liệu trong 1 board. Dữ liệu tối thiểu của board này là 4GB, còn tối đa là 8TB (tùy theo độ mạnh của hệ điều hành)

MySQL cho phép điều chỉnh tùy ý để các lập trình viên có thể thao tác thay đổi cho website của mình.

* Php:

PHP (viết tắt của Hypertext Pre-processor, tiền thân là Personal Home Page) là một ngôn ngữ lập trình kịch bản, được ứng dụng trong các loại website (tĩnh và động) và ứng dụng website. Các script được viết bằng ngôn ngữ lập trình PHP sẽ được các server có cài đặt PHP hiểu được và “tiêu hóa” chúng. Cụ thể, các script này sẽ được lưu dưới dạng tệp đuôi “.php”, sau đó các server sẽ diễn giải chúng và đưa lên website tùy theo template web mà lập trình viên lựa chọn.

Ngôn ngữ lập trình PHP có thể được nhúng trong HTML, được ứng dụng trong quản lý các loại nội dung website (nhất là nội dung động) cũng như quản lý các dữ liệu thống kê liên quan đến website như số phiên, thời gian phiên… PHP được sử dụng nhiều trong các loại Database phổ biến như MySQL, PostgreSQL, Oracle, Sybase, Informix, và Microsoft SQL Server.

PHP thực thi rất tuyệt vời, đặc biệt khi được biên dịch như là một Apache Module trên Unix side. MySQL Server, khi được khởi động, thực thi các truy vấn phức tạp với các tập hợp kết quả khổng lồ trong thời gian Record-setting.

Các chuyên gia code từng đánh giá rằng PHP là một ngôn ngữ lập trình tuyệt vời, có khả năng phiên dịch không khác gì một Apache Module trên Unixside. Đồng thời, chúng còn cho phép hoạt động theo các giao thức nổi tiếng như POP3, IMAP, và LDAP. Ngoài ra, ngôn ngữ lập trình cải tiến từ PHP, PHP4, còn tương hỗ với một ngôn ngữ lập trình khác là Java cùng các cấu trúc đối tượng phân phối như COM và COBRA nhằm vào mục đích thiết lập website cùng ứng dụng website. Đặc biệt hơn nữa, PHP có cú pháp nhập lệnh tương tự với ngôn ngữ lập trình C.

* Perl:

PERL có tên đầy đủ là Practical Extraction and Report Language, là một trong những ngôn ngữ lập trình hiện đại hoạt động song song với PHP trong Xampp có khả năng lọc bỏ những dữ liệu thừa và cung cấp những dữ liệu quan trọng trong việc thiết lập website.

Tuy không nổi bằng PHP, Perl cũng có những ưu điểm nhất định khiến nhiều lập trình viên trên toàn thế giới cảm thấy ấn tượng và ứng dụng cho website của mình.

Cộng đồng sử dụng Perl khá lớn, góp công vào việc xây dựng kho mã lệnh CPAN khổng lồ cho các lập trình viên.

Tương tự như PHP, Perl có cú pháp giống ngôn ngữ lập trình C.

Độ linh hoạt của Perl cao, cho phép lập trình viên có thể tùy biến chúng để giải quyết các vấn đề liên quan đến website và ứng dụng.

# CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

## 2.1 Phân tích hệ thống

### 2.1.1. Sơ đồ phân rã chức năng

|  |
| --- |
|  |

Hình 2 Sơ đồ phân rã chức năng

### 2.1.2. Sơ đồ luồng dữ liệu ngữ cảnh

|  |
| --- |
|  |

Hình 3 Sơ đồ luồng dữ liệu ngữ cảnh

### 2.1.3 Sơ đồ luồng dữ liệu mức đỉnh

|  |
| --- |
|  |

Hình 4 Sơ đồ luồng dữ liệu mức đỉnh

### 2.1.4 Sơ đồ mức dưới đỉnh của Quản lý tài khoản

|  |
| --- |
|  |

Hình 5 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý tài khoản

### 2.1.5. Sơ đồ mức dưới đỉnh của Quản lý sản phẩm

|  |
| --- |
|  |

Hình 6 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý sản phẩm

### 2.1.6. Sơ đồ mức dưới đỉnh của Quản lý hóa đơn và thanh toán

|  |
| --- |
|  |

Hình 7 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý hóa đơn và thanh toán

### 2.1.7. Sơ đồ mức dưới đỉnh của Quản lý phản hồi

|  |
| --- |
|  |

Hình 8 Sơ đồ luồng dữ liệu mức đỉnh của Quản lý phản hồi

## 2.2. Biểu đồ cơ sở dữ liệu

|  |
| --- |
|  |

Hình 10 Biểu đồ cơ sở dữ liệu

## 2.3 Mô tả bảng cơ sở dữ liệu

* Bảng user

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int(100) | Khóa chính | mã |
| name | varchar(100) |  | tên\_ |
| email | varchar(100) |  | email |
| password | varchar(100) |  | mật khẩu |
| user\_type | varchar(20) |  | loại |
| sdt | varchar(11) |  | số điện thoại |
| diachi | varchar(50) |  | địa chỉ |

Bảng 1.Bảng user

* Bảng wishlist

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int(100) | khóa chính | mã |
| user\_id | int(100) | khóa ngoài | loại |
| pid | int(100) |  | pid |
| name | varchar(100) |  | tên |
| price | int(100) |  | giá |
| image | varchar(100) |  | ảnh |

Bảng 2.Bảng wishlist

* Bảng products

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int(100) | khóa chính | mã |
| name | varchar(100) | khóa ngoài | tên |
| details | varchar (500) |  | chi tiết |
| price | int(100) |  | giá |
| image | varchar(100) |  | ảnh |

Bảng 3.Bảng products

* Bảng orders

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int(100) | khóa chính | mã |
| user\_id | int(100) | khóa ngoài | loại |
| name | varchar (100) |  | tên |
| number | varchar(12) |  | số |
| email | varchar(100) |  | email |
| method | varchar(50) |  | phương pháp |
| address | varchar(500) |  | địa chỉ |
| total\_products | varchar(1000) |  | tổng\_sản phẩm |
| total\_price | int(100) |  | tổng\_đơn giá |
| placed\_on | varchar(50) |  | đặt\_trên |
| payment\_status | varchar(20) |  | trạng thái\_thanh toán |

Bảng 4.Bảng oders

* Bảng message

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int(100) | khóa chính | mã |
| user\_id | int(100) | khóa ngoài | loại |
| name | varchar (100) |  | tên |
| email | varchar(100) |  | email |
| number | varchar (12) |  | số |
| message | varchar(50) |  | lời nhắn |

Bảng 5.Bảng message

* Bảng cart

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int(100) | khóa chính | mã |
| user\_id | int(100) | khóa ngoài | loại |
| pid | int (100) |  | pid |
| name | varchar(100) |  | tên |
| price | int (100) |  | giá |
| quantity | int(100) |  | số lượng |
| image | varchar(100) |  | ảnh |

Bảng 6.Bảng cart

## 2.4. Biểu đồ use case tổng quát

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 11 Biểu đồ use case tổng quát

## 2.5. Đặc tả use case

### 2.5.1. Use case đăng nhập

1. Biểu đồ use case:

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 12 Biểu đồ use case đăng nhập

1. Đặc tả use case:

| **Tên UC** | **Đăng nhập** |
| --- | --- |
| **Tác nhân** | Quản trị viên, giảng viên, sinh viên |
| **Mục đích** | Cho phép người dùng truy cập vào hệ thống quản lý sinh viên |
| **Mô tả** | Người dùng đăng nhập vào hệ thống thông qua “tên đăng nhập” và “mật khẩu” |
| **Tiền điều kiện** | Người dùng đã có tài khoản hợp lệ trong hệ thống |

**Luồng sự kiện chính (Luồng thành công)**

1. Người dùng chọn chức năng “Đăng nhập” từ giao diện hệ thống.
2. Hệ thống hiển thị form nhập “tên đăng nhập” và “mật khẩu”.
3. Người dùng nhập thông tin và nhấn “Đăng nhập”.
4. Hệ thống kiểm tra thông tin:
   * Nếu đúng: Hệ thống cho phép truy cập và chuyển đến giao diện tương ứng với vai trò (quản trị viên, giảng viên hoặc sinh viên).
   * Nếu sai: Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.

**Luồng sự kiện phụ (Ngoại lệ)**

* **4.1** Người dùng nhập sai định dạng của “tên đăng nhập” hoặc “mật khẩu” → Hệ thống yêu cầu nhập lại đúng định dạng.
* **4.2** Người dùng nhập thiếu “tên đăng nhập” hoặc “mật khẩu” → Hệ thống yêu cầu nhập đầy đủ thông tin.
* **4.3** Người dùng để trống “tên đăng nhập” hoặc “mật khẩu” → Hệ thống yêu cầu nhập đầy đủ thông tin.
* **4.4** Người dùng nhập sai “tên đăng nhập” hoặc “mật khẩu” (dù đúng định dạng) → Hệ thống thông báo "Đăng nhập thất bại" và yêu cầu thử lại.

1. Biểu đồ trình tự:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 13 Biểu đồ trình tự đăng nhập

1. Biểu đồ hoạt động:



Hình 14 Biểu đồ hoạt động đăng nhập

### 2.5.2. Use case đăng ký

1. Biểu đồ use case:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 15 Biểu đồ use case đăng ký

1. Đặc tả use case:

|  |  |
| --- | --- |
| **Tên UC** | **Đăng kí tài khoản** |
| Tác nhân | Khách xem |
| Mục đích | Đăng kí hệ thống |
| Mô tả | Người dùng đăng ký làm thành viên của hệ thống |
| Tiền điều kiện | Khách xem chưa có tài khoản hệ thống |
| Luồng  sự kiện chính | 1. Người sử dụng chọn chức năng “Tạo tài khoản”. 2. Hệ thống hiển thị form “Đăng kí”. 3. Người sử dụng nhập thông tin vào form “Đăng kí tài khoản”   4. Hệ thống kiểm tra dữ liệu. Nếu đúng thông báo thành công và lưu vào CSDL và chuyển sang form “Đăng nhập”. |
| Luồng  sự kiện phụ | 3.1 Nhập sai định dạng tên đăng nhập, mật khẩu yêu càu nhập lại  3.2 Nhập thiếu thông tin: email, tên đăng nhập, mật khẩu, nhập lại mật khẩu yêu cầu nhập đầy đủ  3.3. Nhập mật khẩu và nhập lại mật khẩu không trùng nhau, yêu cầu nhập lại  4.1 Thông tin tài khoản đã tồn tại trong CSDL, thông báo tài khoản đã tồn tại  4.2 Nhập thiếu thông tin form, yêu cầu nhập đầy đủ  4.3 Bỏ trống thông tin, yêu cầu nhập đầy đủ |

1. Biểu đồ trình tự:

A diagram of a company

AI-generated content may be incorrect.

Hình 16 Biểu đồ trình tự đăng ký

Biểu đồ hoạt động:

Hình 17 Biểu đồ hoạt động đăng ký

### 2.5.3. Use case tìm kiếm

1. Biểu đồ use case:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 18 Biểu đồ use case tìm kiếm

1. Đặc tả use case:

| **Thành phần** | **Nội dung** |
| --- | --- |
| **Tên UC** | Tìm kiếm sinh viên theo tên |
| **Tác nhân** | Cán bộ quản lý, giảng viên, sinh viên xem (nếu hệ thống cho phép truy cập công khai) |
| **Mục đích** | Hỗ trợ người dùng tra cứu thông tin sinh viên một cách nhanh chóng qua tên |
| **Mô tả** | Người dùng nhập tên (hoặc một phần tên) để tìm kiếm sinh viên trong hệ thống quản lý |
| **Tiền điều kiện** | Không cần đăng nhập; hoặc đăng nhập (nếu là cán bộ/giảng viên) |
| **Luồng sự kiện chính** | 1. Người sử dụng chọn chức năng “Tìm kiếm sinh viên” 2. Người sử dụng nhập tên (hoặc từ khóa liên quan đến tên sinh viên) 3. Hệ thống kiểm tra và truy xuất CSDL   Hệ thống hiển thị danh sách sinh viên phù hợp kết quả tìm kiếm |

# CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG

## 3.1. Quá trình triển khai hệ thống

Việc triển khai hệ thống website **quản lý sinh viên** được thực hiện theo quy trình từng bước nhằm đảm bảo tính logic, hiệu quả và độ ổn định trong quá trình phát triển và vận hành. Các bước chính bao gồm:

**Phân tích yêu cầu hệ thống**

Nhóm tiến hành khảo sát thực tế tại các cơ sở giáo dục để tìm hiểu nhu cầu của các đối tượng sử dụng như phòng đào tạo, giảng viên, sinh viên. Từ đó xác định các chức năng cốt lõi của hệ thống bao gồm:

* Quản lý thông tin sinh viên (hồ sơ cá nhân, lớp, khóa, trạng thái học tập)
* Tìm kiếm và lọc sinh viên theo nhiều tiêu chí
* Quản lý điểm học tập và kết quả rèn luyện
* Quản lý lớp học, môn học và thời khóa biểu
* Quản lý đăng ký học phần, học phí
* Quản lý tài khoản người dùng (sinh viên, giảng viên, quản trị viên)

**Yêu cầu phi chức năng** cũng được xác định:

* Tốc độ phản hồi nhanh
* Bảo mật thông tin cá nhân và điểm số
* Khả năng mở rộng cho nhiều khoa/ngành
* Giao diện trực quan, dễ sử dụng trên mọi thiết bị

**Thiết kế hệ thống**

Giai đoạn thiết kế tập trung vào ba nội dung chính:

* **Thiết kế giao diện (UI):** Xây dựng giao diện người dùng với HTML, CSS và Bootstrap. Giao diện được tối ưu để sử dụng dễ dàng với các nhóm người dùng khác nhau như quản trị viên, giảng viên và sinh viên.
* **Thiết kế cơ sở dữ liệu:** Dựa trên mô hình thực thể – quan hệ (ERD), nhóm xây dựng sơ đồ cơ sở dữ liệu với các bảng như: students, courses, classes, grades, users, schedules, payments đảm bảo tính liên kết, mở rộng và bảo trì dễ dàng.
* **Thiết kế luồng xử lý:** Dựng sơ đồ luồng dữ liệu (DFD), sơ đồ use-case và sequence diagram để thể hiện rõ các tương tác như đăng ký môn học, nhập điểm, xem bảng điểm, tra cứu lớp học.

**Lập trình và xây dựng chức năng**

Dựa theo bản thiết kế, nhóm bắt đầu lập trình hệ thống sử dụng:

* **Ngôn ngữ chính:** PHP (backend), HTML/CSS/JS (frontend), SQL (quản lý dữ liệu)
* **Thư viện hỗ trợ:** Bootstrap (giao diện responsive), Font Awesome (biểu tượng), jQuery (tương tác động)

**Các chức năng đã xây dựng bao gồm:**

* Đăng ký – đăng nhập người dùng
* Thêm/sửa/xóa thông tin sinh viên
* Quản lý lớp, khóa, môn học
* Nhập điểm và xem điểm
* Tra cứu sinh viên theo tên/lớp/ngành
* Hệ thống phân quyền người dùng (admin, giảng viên, sinh viên)

**Kiểm thử hệ thống**

Sau khi xây dựng xong, nhóm tiến hành các loại kiểm thử:

* **Kiểm thử chức năng:** Đảm bảo các chức năng hoạt động đúng logic
* **Kiểm thử giao diện:** Kiểm tra UI dễ dùng, không lỗi hiển thị
* **Kiểm thử tương thích:** Đảm bảo hoạt động tốt trên nhiều trình duyệt và thiết bị
* **Kiểm thử dữ liệu:** Kiểm tra độ chính xác trong truy xuất và xử lý thông tin sinh viên

Các lỗi được ghi nhận và khắc phục để đảm bảo hệ thống vận hành ổn định.

**Triển khai cục bộ**

Hệ thống được triển khai thử nghiệm trên **localhost thông qua XAMPP**. Môi trường mô phỏng đầy đủ như máy chủ thực tế, cho phép nhóm phát triển và người dùng thử nghiệm đánh giá hiệu quả của hệ thống.

**Đánh giá và cải tiến**

Sau giai đoạn vận hành thử nghiệm, nhóm tiến hành thu thập phản hồi từ người dùng giả định (giảng viên, cán bộ quản lý, sinh viên) để xác định điểm mạnh – điểm yếu. Từ đó, hệ thống được điều chỉnh về:

* Giao diện: dễ dùng, thuận tiện
* Tốc độ xử lý: nhanh, mượt
* Chức năng: bổ sung tiện ích như thống kê, xuất file, thông báo tự động

Mục tiêu là tối ưu trải nghiệm người dùng và khả năng áp dụng thực tế tại các đơn vị đào tạo.

## 3.2. Giao diện một số hệ thống

A screenshot of a computer

AI-generated content may be incorrect.

Hình 28 Giao diện trang chủ

Giao diện trang đăng nhập được minh họa bằng hình ảnh dưới đây

A login screen with buildings in the background

AI-generated content may be incorrect.

Hình 29 Giao diện trang đăng nhập

Khách hàng có thể đăng ký tài khoản theo hình minh họa dưới đây

A screenshot of a building

AI-generated content may be incorrect.3333

Hình 30 Giao diện trang đăng kí

Tại trang chủ, người dùng có thể thực hiện tìm kiếm sản phẩm theo tên, danh mục

A screenshot of a computer

AI-generated content may be incorrect.

Hình 31 Giao diện chức năng tìm kiếm, đăng ký môn học

Sinh viên có thể chọn học phần, môn học có trong học kỳ để đăng ký

# KẾT LUẬN

Công nghệ thông tin mở ra triển vọng to lớn trong việc đổi mới các phương pháp và hình thức kinh doanh trực tuyến. Những phương pháp tiếp cận mới như dạy học phát hiện và giải quyết vấn đề, cùng với việc ứng dụng công nghệ thông tin trong quản lý kinh doanh, đã càng làm tăng khả năng và hiệu quả của các hoạt động thương mại điện tử. Tiếp thu và ứng dụng những thành tựu của công nghệ thông tin vào công việc kinh doanh, đặc biệt là trong việc quản lý và bán hàng trực tuyến, là rất cần thiết để đạt được hiệu quả quản lý tối ưu.

Trong quá trình thực hiện đồ án, tôi đã học hỏi được nhiều kinh nghiệm thực tiễn. Tôi hiểu sâu hơn các vấn đề liên quan đến phát triển website và đã tiếp thu những kiến thức mới, học hỏi thêm khả năng liên kết các vấn đề để học cách sáng tạo hơn trong công việc. Việc cải thiện hiệu năng của website là vô cùng quan trọng, nhất là khi lượng người dùng và dữ liệu ngày càng tăng. Tôi dự định sẽ tối ưu lại toàn bộ câu truy vấn và thiết kế lại cơ sở dữ liệu để đảm bảo website hoạt động mượt mà và hiệu quả.

Thiết kế trải nghiệm người dùng (UX) cho website vẫn đang là một lĩnh vực đang phát triển mạnh tại Việt Nam và thu hút sự quan tâm lớn từ cộng đồng. Trong thời gian tới, với những kiến thức và kỹ năng đã học hỏi được, tôi sẽ cố gắng rèn luyện và nâng cao năng lực chuyên môn để có thể phát triển công việc thiết kế UX một cách tốt nhất.

Cuối cùng, tôi xin chân thành cảm ơn quý công ty cũng như các thầy cô trong khoa Công nghệ thông tin đã tạo điều kiện và giúp đỡ tôi trong quá trình học tập tại trường. Tôi rất mong nhận được những sự đóng góp và nhận xét từ phía thầy cô để báo cáo của tôi có thể hoàn thiện hơn.

# TÀI LIỆU THAM KHẢO

[1] Trang tài liệu PHP, <https://www.php.net/manual/en/>  
[2] Trang tài liệu MySQL, <https://dev.mysql.com/doc/>  
[3] Hướng dẫn sử dụng XAMPP, https://www.apachefriends.org/index.html  
[4] Trang tài liệu Bootstrap, <https://getbootstrap.com/>  
[5] Hướng dẫn HTML và CSS, <https://www.w3schools.com/>  
[6] Trang tài liệu JavaScript, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
[7] Hướng dẫn sử dụng Font Awesome, <https://fontawesome.com/>  
[8] Hướng dẫn sử dụng Visual Studio Code, <https://code.visualstudio.com/docs>  
[9] Hướng dẫn Git và GitHub, <https://git-scm.com/doc>  
[10] Tài liệu về bảo mật trong PHP, <https://www.php.net/manual/en/security.php>