

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG WEBSITE CÔNG THÔNG TIN CỦA
TRƯỜNG THPT

Người hướng dẫn: **ThS. NGUYỄN THỊ VŨ THẢO**

Sinh viên thực hiện: **PHẠM MINH HIẾU**

Mã số sinh viên: **2151220156**

Lớp: **21CT3**

Đà Nẵng,/202...

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG WEBSITE CÔNG THÔNG TIN CỦA
TRƯỜNG THPT

Người hướng dẫn: **ThS. NGUYỄN THỊ VŨ THẢO**

Sinh viên thực hiện: **PHẠM MINH HIẾU**

Mã số sinh viên: **2151220156**

Lớp: **21CT3.**

Đà Nẵng,/202...

TÓM TẮT

Tên đề tài: xây dựng website Cổng thông tin của trường THPT

Sinh viên thực hiện: Phạm Minh Hiếu

Mã số sinh viên: 2151220156 Lớp: 21CT3

Đề tài xây dựng hệ thống quản lý học sinh cho trường THPT nhằm hỗ trợ giảng dạy, theo dõi kết quả học tập và quản trị thông tin. Hệ thống phân quyền theo ba vai trò: Admin (quản lý người dùng, lớp, môn, phân công, tin tức, báo cáo), Teacher (quản lý lớp phụ trách, nhập điểm, gửi thông báo, thời khóa biểu) và Student (tra cứu thông tin cá nhân, điểm số, thời khóa biểu, nhận thông báo). Ngoài ra, hệ thống có chức năng tìm kiếm, lọc dữ liệu, đảm bảo bảo mật và ghi log thao tác. Cấu trúc đồ án gồm các phần: giới thiệu đề tài, phân tích yêu cầu, thiết kế hệ thống, xây dựng cơ sở dữ liệu, phát triển giao diện và chức năng, kiểm thử và đánh giá kết quả, cuối cùng là phụ lục và tài liệu tham khảo.

Báo cáo gồm 3 chương chính, ngoài phần mở đầu, danh mục và phụ lục. Nội dung từng chương được tóm tắt như sau:

- ❖ Chương 1: Trình bày cơ sở lý thuyết về phát triển web, ngôn ngữ PHP và framework Laravel. Chương này phân tích đặc điểm, ưu điểm của PHP, cách hoạt động trên web server, đồng thời giới thiệu cấu trúc thư mục, mô hình MVC và lợi ích khi sử dụng Laravel. Ngoài ra, chương cũng đề cập đến cơ sở dữ liệu SQLite, đặc điểm, cách lưu trữ và kết nối với PHP/Laravel.
- ❖ Chương 2: Giới thiệu hệ thống quản lý học sinh, mục tiêu, phạm vi và các đối tượng sử dụng. Chương này mô tả các chức năng chính, sơ đồ Use Case, các tác nhân, cùng mô tả chi tiết các Use Case quan trọng. Phần cơ sở dữ liệu trình bày các bảng dữ liệu chính, mối quan hệ, mô hình ERD và thiết kế xử lý thông qua sơ đồ hoạt động, đảm bảo hệ thống logic và dễ triển khai.
- ❖ Chương 3: Trình bày kết quả xây dựng hệ thống thông qua thiết kế giao diện. Bao gồm giao diện dành cho học sinh (tra cứu thông tin, điểm số, thời khóa biểu), giao diện quản trị (quản lý người dùng, lớp, môn, tin tức, báo cáo) và giao diện giáo viên (quản lý lớp phụ trách, nhập điểm, gửi thông báo). Chương này minh họa trực quan các chức năng đã được phân tích và thiết kế ở chương trước.

LỜI CẢM ƠN

Trong quá trình học tập và thực hiện đồ án, em đã nhận được sự quan tâm, giảng dạy và hướng dẫn tận tình từ quý thầy cô trong khoa Công nghệ Thông tin. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn đã luôn hỗ trợ, góp ý và định hướng để em hoàn thành tốt đồ án này.

Em cũng xin cảm ơn nhà trường đã tạo điều kiện thuận lợi về môi trường học tập và nghiên cứu. Mặc dù đã nỗ lực hết sức, nhưng đồ án khó tránh khỏi những thiếu sót. Em kính mong nhận được sự góp ý của quý thầy cô để có thể hoàn thiện hơn trong tương lai.

CAM ĐOAN

Tôi xin cam đoan đồ án tốt nghiệp này là kết quả của quá trình học tập, nghiên cứu và làm việc nghiêm túc của bản thân dưới sự hướng dẫn của giảng viên phụ trách. Các số liệu, kết quả và nội dung trình bày trong đồ án đều trung thực, không sao chép từ bất kỳ nguồn nào mà không trích dẫn rõ ràng.

Tôi hoàn toàn chịu trách nhiệm về tính trung thực và liêm chính học thuật của đồ án này. Nếu phát hiện có bất kỳ hành vi gian lận, sao chép hoặc vi phạm quy định về đạo đức học thuật, tôi xin hoàn toàn chịu trách nhiệm trước nhà trường và pháp luật.

Tôi xin chân thành cảm ơn sự hướng dẫn tận tình của giảng viên và sự hỗ trợ từ các thầy cô, bạn bè trong suốt quá trình thực hiện đồ án.

Sinh viên thực hiện

{Chữ ký, họ và tên sinh viên}

MỤC LỤC

Tóm tắt.....	1
Lời cảm ơn.....	i1
Cam đoan.....	1ii
Mục lục	vi
Danh sách các bảng biểu, hình vẽ	v
Mở đầu.....	vii
CHƯƠNG 1. CÁC LÝ THUYẾT LIÊN QUAN	1
1.1. Giới thiệu về phát triển web.....	1
1.2. Tổng quan về ngôn ngữ php	2
1.2.1. Đặt điểm và ưu điểm của PHP	3
1.2.2. Cách PHP hoạt động trên web server.....	3
1.3. Giới thiệu về Framework Laravel.....	4
1.3.1. Cấu trúc thư mục của Laravel	5
1.3.2. Mô hình MVC trong Laravel	6
1.3.3. Ưu điểm khi sử dụng Laravel.....	7
1.4. Tổng quan về cơ sở dữ liệu SQLite	8
1.4.1. Giới thiệu và đặc điểm của SQLite	8
1.4.2. Cấu trúc và cách lưu trữ dữ liệu.....	9
1.4.3. Kết nối SQLite với PHP và Laravel.....	11
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13
2.1. Phân tích chức năng	13
2.1.1. Giới thiệu hệ thống.....	13
2.1.2. Chức năng quản lý hệ thống (Admin).....	14
2.1.3. Chức năng dành cho giáo viên (Teacher)	14
2.1.4. Chức năng dành cho học sinh (Student)	14

2.2. Sơ đồ Use Case	15
2.2.1. Các tác nhân (Actor)	15
2.2.2. Biểu đồ Use Case tổng quát	16
2.2.3 Mô tả các Use Case chính	17
2.3. Sơ đồ hoạt động	19
2.3.1. Admin - Quản lý tài khoản.....	19
2.3.2. Teacher - Nhập điểm học sinh	20
2.3.3. Student - Xem điểm học tập.....	21
2.4. Cơ sở dữ liệu (CSDL)	19
2.4.1. Các bảng dữ liệu chính.....	19
2.4.2. Mối quan hệ giữa các bảng	20
2.4.3. Mô hình ERD	22
CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM	25
3.1. Kết quả giao diện	25
3.1.1. Giao diện học sinh.....	26
3.1.2. Giao diện quản trị.....	28
3.1.3. Giao diện giáo viên	30
KẾT LUẬN	32
TÀI LIỆU THAM KHẢO	33
PHỤ LỤC	

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Danh sách các hình vẽ

Hình 1.1: Tổng quan PHP	2
Hình 1.2: Giới thiệu về framework Laravel.....	4
Hình 1.3: Tổng quan về sqlite	8
Hình 2.1: Use case tổng quát	16
Hình 2.2: Sơ đồ hoạt động - Admin.....	19
Hình 2.3: Sơ đồ hoạt động - Teacher	20
Hình 2.4: Sơ đồ hoạt động - Student.....	21
Hình 2.5: Sơ đồ erd	25
Hình 3.1: Chức năng xem thông tin cá nhân(Student).....	26
Hình 3.2: Chức năng xem điểm(Student)	27
Hình 3.3: Chức năng quản lý danh sách học sinh(Admin)	28
Hình 3.4: Chức năng phân công giáo viên(Admin)	29
Hình 3.5: Chức năng nhập điểm(Teacher).....	30
Hình 3.6: Chức năng xem danh sách học sinh(teacher).....	31

Danh sách các bảng

Bảng 2.1: Mô tả các use case	17
Bảng 2.2: Các bảng dữ liệu chính	22

MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc xây dựng một website Cổng thông tin cho trường THPT không chỉ giúp nhà trường quản lý hiệu quả mà còn tạo ra môi trường học tập hiện đại, thuận tiện cho học sinh. Đề tài “Xây dựng website Cổng thông tin của trường THPT” được thực hiện với trọng tâm là phục vụ nhu cầu tra cứu, tiếp nhận thông tin và tương tác của học sinh trong môi trường học đường.

Website được thiết kế nhằm hỗ trợ học sinh tra cứu thời khóa biểu, điểm số, thông báo từ giáo viên, tin tức từ nhà trường, cũng như cập nhật hồ sơ cá nhân. Ngoài ra, hệ thống còn giúp học sinh tiếp cận thông tin học tập một cách nhanh chóng, chính xác và bảo mật. Các chức năng dành cho giáo viên và quản trị viên được xây dựng để hỗ trợ việc cung cấp và quản lý thông tin cho học sinh.

Phạm vi nghiên cứu của đề tài tập trung vào các chức năng phục vụ học sinh trong trường THPT, bao gồm quản lý thông tin cá nhân, điểm số, thời khóa biểu, thông báo và tin tức. Đối tượng nghiên cứu chính là học sinh, với sự hỗ trợ từ giáo viên và quản trị viên trong việc cung cấp dữ liệu và quản lý hệ thống.

Phương pháp nghiên cứu bao gồm khảo sát nhu cầu thực tế của học sinh, phân tích yêu cầu sử dụng, thiết kế hệ thống theo mô hình UML, xây dựng cơ sở dữ liệu quan hệ và phát triển giao diện web thân thiện. Các công cụ sử dụng gồm phần mềm thiết kế sơ đồ, hệ quản trị cơ sở dữ liệu và ngôn ngữ lập trình web như HTML, CSS, JavaScript và PHP.

CHƯƠNG 1: CÁC LÝ THUYẾT LIÊN QUAN

1.1. Giới thiệu về phát triển web

Phát triển web là quá trình xây dựng, triển khai và duy trì các trang web hoặc ứng dụng web nhằm đáp ứng nhu cầu sử dụng của người dùng trên Internet. Đây là một lĩnh vực quan trọng trong kỷ nguyên số, đóng vai trò nền tảng trong việc cung cấp thông tin, dịch vụ và trải nghiệm tương tác trực tuyến. Quá trình phát triển web không chỉ đơn thuần là thiết kế giao diện mà còn bao gồm nhiều hoạt động kỹ thuật phức tạp như lập trình chức năng, xử lý cơ sở dữ liệu, tối ưu hóa hiệu suất, đảm bảo bảo mật và khả năng mở rộng của hệ thống.

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin, các trang web đã vượt xa vai trò truyền thống là nơi hiển thị thông tin tĩnh. Thay vào đó, chúng trở thành công cụ tương tác mạnh mẽ giữa người dùng và hệ thống, hỗ trợ các hoạt động như thương mại điện tử, quản lý dữ liệu, truyền thông xã hội, giáo dục trực tuyến và nhiều lĩnh vực khác. Các ứng dụng web hiện đại thường tích hợp nhiều công nghệ tiên tiến như trí tuệ nhân tạo (AI), học máy (Machine Learning), điện toán đám mây (Cloud Computing), và các giao thức bảo mật nâng cao nhằm nâng cao trải nghiệm người dùng và đảm bảo an toàn thông tin.

Phát triển web hiện đại thường được chia thành hai mảng chính:

Front-end Development: Là phần xử lý giao diện và trải nghiệm người dùng. Front-end developer chịu trách nhiệm thiết kế bố cục, màu sắc, kiểu chữ, hiệu ứng và các thành phần tương tác trên trình duyệt. Các công nghệ phổ biến bao gồm HTML, CSS, JavaScript, cùng với các thư viện và framework như React, Angular, Vue.js,...

Back-end Development: Là phần xử lý logic nghiệp vụ, quản lý dữ liệu và giao tiếp với máy chủ. Back-end developer xây dựng các chức năng như đăng nhập, xử lý biểu mẫu, lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu. Các ngôn ngữ và công nghệ thường dùng gồm PHP, Python, Java, Node.js, cùng với các framework như Laravel, Django, Express,...

Ngoài ra, còn có vai trò Full-stack Developer, là người có khả năng đảm nhiệm cả front-end và back-end, giúp tối ưu hóa quy trình phát triển và triển khai hệ thống.

Phát triển web không chỉ đòi hỏi kiến thức kỹ thuật mà còn yêu cầu khả năng phân tích nhu cầu người dùng, tư duy thiết kế, kỹ năng làm việc nhóm và khả năng cập nhật công nghệ mới. Đây là một lĩnh vực năng động, luôn thay đổi và mở ra nhiều cơ hội nghề nghiệp hấp dẫn cho những ai đam mê công nghệ và sáng tạo.

1.2. Tổng quan về ngôn ngữ PHP



Hình 1.1 Tổng quan PHP

PHP (viết tắt của “Hypertext Preprocessor”) là một ngôn ngữ lập trình kịch bản phía máy chủ, mã nguồn mở, được thiết kế chuyên biệt cho phát triển web. Ra đời vào năm 1994 bởi Rasmus Lerdorf, PHP đã nhanh chóng trở thành một trong những ngôn ngữ phổ biến nhất trong lĩnh vực xây dựng ứng dụng web nhờ tính linh hoạt, dễ sử dụng và khả năng tích hợp mạnh mẽ với các công nghệ web hiện đại.

Một trong những điểm nổi bật của PHP là khả năng nhúng trực tiếp vào mã HTML, cho phép lập trình viên dễ dàng tạo ra các trang web động mà không cần sử dụng các công cụ phức tạp. PHP được thực thi trên máy chủ, nghĩa là mã PHP được xử lý trước khi gửi kết quả HTML đến trình duyệt của người dùng. Điều này giúp tăng tính bảo mật và hiệu suất xử lý.

PHP thường được sử dụng để xây dựng các hệ thống quản lý nội dung (CMS), cổng thông tin điện tử, thương mại điện tử, diễn đàn, blog và các ứng dụng web tương tác. Nhiều nền tảng nổi tiếng như WordPress, Joomla, Drupal đều được phát triển dựa trên PHP.

1.2.1. Đặc điểm và ưu điểm của PHP

Mã nguồn mở và miễn phí: PHP được phát hành dưới giấy phép mã nguồn mở, cho phép người dùng sử dụng, chỉnh sửa và phân phối mà không phải trả phí. Cộng đồng PHP rất lớn và năng động, cung cấp nhiều tài nguyên, thư viện và công cụ hỗ trợ.

Dễ học và dễ sử dụng: Cú pháp của PHP đơn giản, gần gũi với các ngôn ngữ như C, Java, giúp người mới dễ tiếp cận và học nhanh. Việc triển khai một ứng dụng web cơ bản với PHP không đòi hỏi kiến thức quá sâu về lập trình.

Tích hợp tốt với HTML: PHP có thể được nhúng trực tiếp vào các tệp HTML, giúp việc xây dựng giao diện và xử lý logic trở nên liền mạch. Điều này đặc biệt hữu ích trong việc tạo ra các trang web động.

Tương thích với nhiều hệ quản trị cơ sở dữ liệu: PHP hỗ trợ kết nối với nhiều hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL, SQLite, Oracle,... giúp lập trình viên linh hoạt trong việc lựa chọn nền tảng lưu trữ dữ liệu.

Hiệu suất cao: PHP có khả năng xử lý nhanh, tiêu tốn ít tài nguyên hệ thống, phù hợp với các ứng dụng vừa và nhỏ. Với sự hỗ trợ của các công cụ như opcode cache (ví dụ: OPcache), hiệu suất của PHP có thể được cải thiện đáng kể.

Hỗ trợ đa nền tảng: PHP có thể chạy tốt trên nhiều hệ điều hành như Windows, Linux, macOS,... và dễ dàng tích hợp với các máy chủ web phổ biến như Apache, Nginx, IIS.

Khả năng mở rộng và bảo trì: PHP hỗ trợ lập trình hướng đối tượng, giúp xây dựng các ứng dụng có cấu trúc rõ ràng, dễ mở rộng và bảo trì. Ngoài ra, còn có nhiều framework mạnh mẽ như Laravel, Symfony, CodeIgniter,... giúp tăng tốc độ phát triển và đảm bảo tính bảo mật.

1.2.2. Cách PHP hoạt động trên Web Server

PHP là ngôn ngữ lập trình phía máy chủ, nghĩa là mã PHP được xử lý tại máy chủ trước khi gửi kết quả đến trình duyệt của người dùng. Quá trình hoạt động của PHP trên Web Server diễn ra theo các bước cơ bản sau:

Người dùng gửi yêu cầu (request)

Khi người dùng truy cập một trang web có sử dụng PHP, trình duyệt sẽ gửi yêu cầu HTTP đến máy chủ web thông qua địa chỉ URL.

Máy chủ web tiếp nhận yêu cầu

Máy chủ web (thường là Apache, Nginx hoặc IIS) sẽ tiếp nhận yêu cầu và xác định tệp PHP cần xử lý. Sau đó, máy chủ chuyển yêu cầu này đến bộ xử lý PHP (PHP interpreter).

PHP thực thi mã lệnh

Bộ xử lý PHP sẽ thực thi các đoạn mã PHP trong tệp. Trong quá trình này, PHP có thể:

- Truy xuất cơ sở dữ liệu (MySQL, PostgreSQL, v.v.)
- Xử lý dữ liệu đầu vào từ người dùng (form, URL, cookie,...)
- Thực hiện các thao tác tính toán, kiểm tra điều kiện, xử lý logic nghiệp vụ

Tạo nội dung HTML động

Sau khi xử lý xong, PHP sẽ tạo ra nội dung HTML (hoặc JSON, XML tùy ứng dụng) để gửi trả về trình duyệt. Người dùng không thấy mã PHP, chỉ thấy kết quả đã được xử lý.

Trình duyệt hiển thị kết quả

Trình duyệt nhận nội dung HTML từ máy chủ và hiển thị giao diện trang web cho người dùng. Quá trình này diễn ra rất nhanh, thường chỉ trong vài mili giây.

1.3. Giới thiệu về Framework Laravel



Hình 1.2 Giới thiệu về framework Laravel

Laravel là một framework PHP mã nguồn mở được phát triển bởi Taylor Otwell từ năm 2011. Laravel được xây dựng nhằm đơn giản hóa quá trình phát triển ứng dụng web bằng cách cung cấp một bộ công cụ mạnh mẽ, cú pháp rõ ràng và cấu trúc tổ chức hợp lý. Framework này tuân theo mô hình kiến trúc MVC (Model - View - Controller), giúp tách biệt rõ ràng giữa phần xử lý dữ liệu, giao diện và điều khiển luồng xử lý, từ đó nâng cao khả năng bảo trì, mở rộng và tái sử dụng mã nguồn.

Laravel nổi bật với các tính năng hiện đại như hệ thống định tuyến linh hoạt, cơ chế ORM (Eloquent) mạnh mẽ, hệ thống xác thực bảo mật, hỗ trợ migration cơ sở dữ liệu, hệ thống template Blade, và khả năng tích hợp với các dịch vụ bên ngoài thông qua API. Laravel cũng hỗ trợ tốt việc phát triển RESTful API, phù hợp với xu hướng phát triển ứng dụng web hiện đại.

Với cộng đồng phát triển lớn mạnh, tài liệu phong phú và hệ sinh thái đa dạng (Laravel Mix, Laravel Sanctum, Laravel Jetstream,...), Laravel đã trở thành một trong những framework PHP phổ biến nhất hiện nay, được sử dụng rộng rãi trong các dự án từ nhỏ đến lớn.

1.3.1. Cấu trúc thư mục của Laravel

Khi khởi tạo một dự án Laravel, hệ thống sẽ tạo ra một cấu trúc thư mục rõ ràng, mỗi thư mục đảm nhiệm một vai trò riêng biệt trong quá trình phát triển ứng dụng. Dưới đây là mô tả các thư mục chính:

- ❖ **app/**: Chứa mã nguồn chính của ứng dụng, bao gồm các thành phần như:
 - **Models**: đại diện cho các bảng dữ liệu.
 - **Controllers**: xử lý logic điều khiển.
 - **Middleware**: xử lý các yêu cầu trước khi đến controller.
- ❖ **bootstrap/**: Chứa các tệp khởi tạo ứng dụng, bao gồm file `app.php` và thư mục `cache/` để lưu trữ các tệp đã biên dịch.
- ❖ **config/**: Lưu trữ các tệp cấu hình cho toàn bộ ứng dụng như cấu hình cơ sở dữ liệu, mail, session, cache, app,...
- ❖ **database/**: Quản lý các thành phần liên quan đến cơ sở dữ liệu:
 - **migrations/**: định nghĩa cấu trúc bảng.
 - **seeders/**: tạo dữ liệu mẫu.
 - **factories/**: tạo dữ liệu giả lập để kiểm thử.
- ❖ **public/**: Là thư mục gốc được truy cập từ trình duyệt. Chứa file `index.php` (điểm khởi đầu của ứng dụng) và các tài nguyên tĩnh như CSS, JavaScript, hình ảnh,...
 - **web.php**: định nghĩa route cho giao diện web.
 - **lang/**: các tệp ngôn ngữ hỗ trợ đa ngôn ngữ.
 - **sass/** và **js/**: mã nguồn CSS và JavaScript.
- ❖ **routes/**: Định nghĩa các tuyến đường (routes) của ứng dụng, liên kết URL với controller tương ứng. Các tệp phổ biến gồm:
 - **web.php**: định nghĩa route cho giao diện web.
 - **api.php**: định nghĩa route cho API.

- ❖ **storage/**: Lưu trữ các tệp tạm, log, file upload và cache. Gồm các thư mục con như:
 - **app/**: lưu trữ file người dùng tải lên.
 - **logs/**: lưu nhật ký hệ thống.
 - **framework/**: lưu cache, session,...
- ❖ **vendor/**: Chứa các thư viện và gói phụ thuộc được cài đặt thông qua Composer. Đây là nơi Laravel và các package bên thứ ba được lưu trữ.

1.3.2. Mô hình MVC trong Laravel

Laravel áp dụng mô hình kiến trúc **MVC (Model – View – Controller)**, giúp tách biệt dữ liệu, giao diện và xử lý logic. Nhờ đó, mã nguồn rõ ràng, dễ bảo trì, mở rộng và hỗ trợ làm việc nhóm hiệu quả.

- ❖ **Model (M)**: Nằm trong thư mục **app/Models**, đại diện cho các bảng dữ liệu trong cơ sở dữ liệu và được xây dựng dựa trên ORM Eloquent. Model định nghĩa thuộc tính, quan hệ (one-to-many, many-to-many,...) và các phương thức xử lý dữ liệu như thêm, sửa, xóa, truy vấn.
- ❖ **View (V)**: Nằm trong thư mục **resources/views**, chứa các file giao diện sử dụng **Blade Template Engine**. View hiển thị dữ liệu cho người dùng, hỗ trợ nhúng PHP vào HTML, điều kiện, vòng lặp, kế thừa layout và tái sử dụng thành phần giao diện. Các tài nguyên tĩnh như CSS, JS, hình ảnh được đặt trong thư mục **public/**.
- ❖ **Controller (C)**: Nằm trong thư mục **app/Http/Controllers**, đóng vai trò trung gian giữa Model và View. Controller tiếp nhận yêu cầu từ người dùng thông qua **routes/**, xử lý logic nghiệp vụ, tương tác với Model để lấy hoặc cập nhật dữ liệu, sau đó trả kết quả về View. Ví dụ: UserController, ProductController.

Lợi ích của mô hình MVC trong Laravel:

- Tách biệt rõ ràng các thành phần, dễ quản lý và mở rộng.
- Tăng khả năng tái sử dụng mã nguồn.
- Hỗ trợ làm việc nhóm hiệu quả.
- Dễ kiểm thử và gỡ lỗi nhờ các thành phần độc lập.

1.3.3. Ưu điểm khi sử dụng Laravel

Laravel là một trong những framework PHP phổ biến nhất hiện nay, được đánh giá cao nhờ vào sự tiện lợi, linh hoạt và khả năng mở rộng. Việc sử dụng Laravel trong phát triển ứng dụng web mang lại nhiều lợi ích nổi bật như sau:

Cú pháp rõ ràng, dễ đọc và dễ bảo trì

Laravel được thiết kế với cú pháp ngắn gọn, dễ hiểu và gần gũi với lập trình viên. Điều này giúp giảm thiểu thời gian viết mã, tăng hiệu suất làm việc và dễ dàng bảo trì, mở rộng hệ thống trong tương lai.

Tích hợp sẵn Eloquent ORM mạnh mẽ

Laravel sử dụng Eloquent ORM (Object-Relational Mapping) để thao tác với cơ sở dữ liệu một cách trực quan và hiệu quả. Eloquent cho phép lập trình viên làm việc với dữ liệu như các đối tượng, hỗ trợ các truy vấn phức tạp mà không cần viết nhiều câu lệnh SQL thuần.

Hệ thống Routing linh hoạt và dễ sử dụng

Laravel cung cấp cơ chế định tuyến (Routing) rõ ràng, cho phép ánh xạ các URL đến các controller hoặc closure một cách dễ dàng. Điều này giúp tổ chức mã nguồn hợp lý và kiểm soát luồng xử lý hiệu quả.

Hỗ trợ Middleware và các tính năng nâng cao

Laravel hỗ trợ Middleware để xử lý các yêu cầu HTTP trước khi đến controller, giúp kiểm soát truy cập, xác thực người dùng, xử lý CORS,... Ngoài ra, Laravel còn tích hợp sẵn các tính năng như:

- Authentication: xác thực người dùng nhanh chóng và an toàn.
- Session: quản lý phiên làm việc của người dùng.
- Queue: xử lý các tác vụ nền như gửi email, xử lý ảnh.
- Mail: hỗ trợ gửi email dễ dàng thông qua nhiều dịch vụ như SMTP, Mailgun, Sendmail,...

Bảo mật cao

- Laravel tích hợp nhiều cơ chế bảo mật giúp bảo vệ ứng dụng khỏi các lỗ hổng phổ biến như:
- SQL Injection: thông qua cơ chế binding dữ liệu trong Eloquent.
- CSRF (Cross-Site Request Forgery): sử dụng token để xác thực các yêu cầu từ form.
- XSS (Cross-Site Scripting): tự động mã hóa dữ liệu đầu ra để ngăn chặn mã độc.

Cộng đồng lớn và tài liệu phong phú

Laravel có một cộng đồng phát triển rộng khắp và năng động. Người dùng có thể dễ dàng tìm thấy tài liệu chính thức, hướng dẫn chi tiết, khóa học, diễn đàn hỗ trợ và hàng ngàn package mở rộng trên Packagist để tích hợp thêm tính năng cho ứng dụng.

1.4. Tổng quan về cơ sở dữ liệu SQLite



Hình 1.3 Tổng quan về sqlite

1.4.1. Giới thiệu và đặc điểm của SQLite

SQLite là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nhẹ, mã nguồn mở, được thiết kế để tích hợp trực tiếp vào các ứng dụng mà không cần cài đặt máy chủ riêng biệt. Khác với các hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL hay Oracle, SQLite hoạt động theo mô hình **serverless**, nghĩa là toàn bộ dữ liệu được lưu trữ trong một tệp duy nhất trên hệ thống tập tin. Điều này giúp việc triển khai và quản lý cơ sở dữ liệu trở nên đơn giản và thuận tiện, đặc biệt phù hợp với các ứng dụng nhỏ, ứng dụng di động hoặc các hệ thống nhúng.

SQLite được sử dụng rộng rãi trong nhiều nền tảng như Android, iOS, trình duyệt web (Firefox, Chrome), phần mềm desktop và các thiết bị IoT. Với khả năng hoạt động ổn định, tốc độ truy xuất nhanh và dung lượng nhỏ gọn, SQLite là lựa chọn lý tưởng cho các ứng dụng không yêu cầu xử lý dữ liệu phức tạp hoặc quy mô lớn.

Đặc điểm nổi bật của SQLite:

Nhẹ, gọn, dễ tích hợp

SQLite có kích thước nhỏ, không yêu cầu cài đặt phức tạp, dễ dàng tích hợp vào các ứng dụng mà không cần cấu hình máy chủ.

Không cần cài đặt riêng biệt

Toàn bộ hệ thống cơ sở dữ liệu được đóng gói trong một thư viện duy nhất, giúp giảm thiểu chi phí triển khai và bảo trì.

Hỗ trợ cú pháp SQL chuẩn

SQLite tuân thủ phần lớn chuẩn SQL-92, cho phép thực hiện các thao tác như SELECT, INSERT, UPDATE, DELETE, JOIN,... giống như các hệ quản trị cơ sở dữ liệu khác.

Tốc độ truy xuất nhanh

Với cơ chế lưu trữ trực tiếp trên tệp tin, SQLite có khả năng truy xuất dữ liệu nhanh chóng, đặc biệt hiệu quả với các ứng dụng có lượng dữ liệu vừa phải.

Thích hợp cho ứng dụng thử nghiệm, di động hoặc hệ thống nhúng

Do không yêu cầu máy chủ, SQLite rất phù hợp để phát triển ứng dụng mẫu, ứng dụng di động, phần mềm desktop hoặc các thiết bị nhúng như Raspberry Pi.

1.4.2. Cấu trúc và cách lưu trữ dữ liệu

SQLite sử dụng một kiến trúc lưu trữ đơn giản nhưng hiệu quả, phù hợp với các ứng dụng nhỏ, di động hoặc hệ thống nhúng. Toàn bộ cơ sở dữ liệu trong SQLite — bao gồm bảng (table), chỉ mục (index), trigger, view và các đối tượng khác — được lưu trữ trong một tệp duy nhất trên hệ thống tệp tin. Tệp này thường có phần mở rộng là .db, .sqlite, hoặc .sqlite3.

Cấu trúc lưu trữ:

Tập cơ sở dữ liệu duy nhất

Không giống như các hệ quản trị cơ sở dữ liệu như MySQL hay PostgreSQL yêu cầu cài đặt máy chủ và lưu trữ dữ liệu phân tán, SQLite gói toàn bộ dữ liệu vào một tệp duy nhất. Điều này giúp việc sao chép, di chuyển và sao lưu cơ sở dữ liệu trở nên đơn giản.

Tổ chức dữ liệu theo hàng và cột

Các bảng trong SQLite được tổ chức theo dạng hàng (row) và cột (column), tương tự như các hệ quản trị cơ sở dữ liệu quan hệ khác. Mỗi bảng có thể chứa nhiều bản ghi, và mỗi bản ghi là một hàng dữ liệu với các trường tương ứng.

Hỗ trợ các đối tượng cơ sở dữ liệu

SQLite hỗ trợ đầy đủ các đối tượng như:

- Table: lưu trữ dữ liệu chính.
- Index: tăng tốc độ truy vấn.
- View: tạo các bảng ảo từ truy vấn SQL.
- Trigger: thực hiện hành động tự động khi có thay đổi dữ liệu.
- Schema: định nghĩa cấu trúc tổng thể của cơ sở dữ liệu.

Cơ chế lưu trữ theo trang (page-based)

SQLite lưu dữ liệu theo từng trang (page), giúp tối ưu hóa hiệu suất đọc/ghi và giảm thiểu rủi ro mất dữ liệu. Mỗi trang có kích thước cố định (thường là 4KB), và các thao tác ghi dữ liệu được thực hiện theo cơ chế giao dịch (transaction) để đảm bảo tính toàn vẹn.

Ưu điểm của cách lưu trữ này:

- Đơn giản hóa triển khai: chỉ cần một tệp duy nhất, không cần cấu hình máy chủ.
- Dễ sao lưu và phục hồi: chỉ cần sao chép tệp .db là có thể phục hồi toàn bộ dữ liệu.
- Hiệu suất cao với dữ liệu vừa phải: truy xuất nhanh, phù hợp với ứng dụng không yêu cầu xử lý dữ liệu lớn.
- Tính di động cao: dễ tích hợp vào ứng dụng di động, phần mềm desktop hoặc thiết bị nhúng.

1.4.3. Kết nối SQLite với PHP và Laravel

Kết nối SQLite với PHP thuần

Trong PHP thuần, việc kết nối với cơ sở dữ liệu SQLite có thể thực hiện thông qua hai cách phổ biến:

- ❖ **Sử dụng lớp SQLite3:** PHP cung cấp lớp SQLite3 tích hợp sẵn để làm việc với cơ sở dữ liệu SQLite. Lập trình viên có thể khởi tạo kết nối, thực hiện truy vấn và xử lý kết quả một cách trực tiếp.

Ví dụ:

```
$db = new SQLite3('database.sqlite');  
// Tạo đối tượng SQLite3 và kết nối đến file database.sqlite  
$result = $db->query('SELECT * FROM users');  
// Thực hiện truy vấn SQL để lấy toàn bộ dữ liệu từ bảng users
```

- ❖ **Sử dụng thư viện PDO (PHP Data Objects) :** PDO là một lớp trừu tượng hóa truy cập cơ sở dữ liệu, hỗ trợ nhiều hệ quản trị khác nhau, trong đó có SQLite. Việc sử dụng PDO giúp mã nguồn linh hoạt và dễ bảo trì hơn.

Ví dụ:

```
$pdo = new PDO('sqlite:database.sqlite');  
// Khởi tạo kết nối PDO đến file database.sqlite  
$stmt = $pdo->query('SELECT * FROM users');  
// Thực hiện truy vấn SQL để lấy dữ liệu từ bảng users
```

Cả hai phương pháp đều hỗ trợ đầy đủ các thao tác CRUD (Create, Read, Update, Delete) và có thể sử dụng trong các ứng dụng nhỏ hoặc thử nghiệm.

Kết nối SQLite với Laravel

Laravel hỗ trợ kết nối với SQLite một cách đơn giản và hiệu quả thông qua cấu hình môi trường. Quá trình thiết lập bao gồm các bước sau:

- ❖ **Tạo file cơ sở dữ liệu SQLite**

Tạo một tệp có phần mở rộng .sqlite hoặc .db, ví dụ: database.sqlite, và đặt trong thư mục database/ của dự án Laravel.

- ❖ **Cấu hình file .env**

Mở file .env trong thư mục gốc của dự án và chỉnh sửa các dòng sau:

```
DB_CONNECTION=sqlite  
DB_DATABASE=/absolute/path/to/database/database.sqlite
```

Giải thích:

DB_CONNECTION=sqlite → Chỉ định loại cơ sở dữ liệu là SQLite.

DB_DATABASE=... → Khai báo đường dẫn tuyệt đối đến file cơ sở dữ liệu SQLite.

Lưu ý: Đường dẫn phải là đường dẫn tuyệt đối đến tệp .sqlite.

Laravel tự động kết nối

Sau khi cấu hình, Laravel sẽ tự động kết nối với cơ sở dữ liệu SQLite thông qua hệ thống ORM Eloquent. Lập trình viên có thể sử dụng các Model để thao tác dữ liệu mà không cần viết câu lệnh SQL trực tiếp.

Ví dụ:

```
$users = User::all();  
// Lấy toàn bộ dữ liệu từ bảng users thông qua Model User  
$user = User::find(1);  
// Tìm một bản ghi có id = 1 trong bảng users  
User::create(['name' => 'Kani', 'email' => 'kani@example.com']);  
// Thêm một bản ghi mới vào bảng users
```

Giải thích:

User::all() → Truy vấn tất cả dữ liệu từ bảng users.

User::find(1) → Lấy dữ liệu của người dùng có id = 1.

User::create([...]) → Thêm dữ liệu mới vào bảng users thông qua Model.

Ngoài ra, Laravel cũng hỗ trợ Migration, Seeder và Factory với SQLite, giúp quá trình phát triển và kiểm thử trở nên thuận tiện hơn.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích chức năng

2.1.1. Giới thiệu hệ thống

Hệ thống dự kiến xây dựng dưới dạng **website cổng thông tin điện tử** dành cho trường Trung học Phổ thông (THPT), nhằm số hóa quy trình quản lý giáo dục và hỗ trợ các bên liên quan (nhà trường, giáo viên, học sinh, phụ huynh) trong việc truy cập, tra cứu và tương tác thông tin.

Mục tiêu và phạm vi

Mục tiêu chính của hệ thống

Hệ thống được xây dựng nhằm đáp ứng các nhu cầu quản lý và truyền thông nội bộ của trường Trung học Phổ thông, với các mục tiêu cụ thể như sau:

Cung cấp cổng thông tin chính thức cho nhà trường

Hệ thống đóng vai trò là nền tảng truyền thông chính thức, cung cấp các thông tin quan trọng như:

- Tin tức hoạt động của trường
- Thông báo học vụ, lịch thi, lịch học
- Thông tin liên hệ, hướng dẫn dành cho học sinh và phụ huynh

Quản lý nội bộ hiệu quả

Hệ thống hỗ trợ quản lý toàn diện các đối tượng và dữ liệu trong nhà trường, bao gồm:

- Hồ sơ học sinh: thông tin cá nhân, lớp học, điểm số, kết quả học tập
- Hồ sơ giáo viên: thông tin giảng dạy, phân công lớp, lịch giảng dạy
- Quản lý lớp học: danh sách học sinh, phân công giáo viên
- Quản lý điểm số và thời khóa biểu: nhập điểm, tra cứu điểm, hiển thị lịch học

Phân quyền rõ ràng theo vai trò người dùng

Hệ thống thiết kế cơ chế phân quyền hợp lý, đảm bảo bảo mật và hiệu quả sử dụng:

- Admin: có toàn quyền quản trị hệ thống, bao gồm quản lý người dùng, dữ liệu, cấu hình hệ thống.
- Teacher: có quyền quản lý lớp học, nhập điểm, gửi thông báo đến học sinh.
- Student: có quyền xem thông tin cá nhân, điểm số, thời khóa biểu và nhận thông báo từ giáo viên hoặc nhà trường.

Ứng dụng dùng nội bộ cho trường THPT

Hệ thống được triển khai trong phạm vi nội bộ của trường, phục vụ công tác quản lý và tra cứu thông tin của giáo viên, học sinh và ban giám hiệu.

Trang công khai dành cho học sinh và phụ huynh

Một phần của hệ thống được thiết kế để công khai thông tin như tin tức, thông báo, lịch hoạt động,... giúp phụ huynh và học sinh dễ dàng theo dõi hoạt động của trường.

Giới hạn phạm vi chức năng

Trong giai đoạn hiện tại, hệ thống không bao gồm các chức năng sau:

- Thanh toán học phí hoặc các khoản thu
- Hệ thống điểm danh tự động bằng mã QR hoặc sinh trắc học
- Tích hợp với các nền tảng bên thứ ba như Google Classroom, Zalo, hoặc hệ thống SMS

Tuy nhiên, các chức năng trên có thể được mở rộng trong tương lai tùy theo nhu cầu thực tế và khả năng phát triển của hệ thống.

2.1.2. Chức năng quản lý hệ thống (Admin)

- Quản lý người dùng: Tạo và phân quyền cho giáo viên, học sinh, đảm bảo truy cập phù hợp theo vai trò.
- Quản lý lớp học và môn học: Tổ chức, cập nhật thông tin lớp, phân công giáo viên giảng dạy.
- Quản lý tin tức và thông báo: Đăng tải các thông tin chính thức của nhà trường lên trang công khai.
- Báo cáo và thống kê: Tổng hợp dữ liệu phục vụ công tác quản lý, đánh giá và ra quyết định.

2.1.3. Chức năng dành cho giáo viên (Teacher)

- Quản lý lớp học phụ trách: Xem danh sách học sinh, theo dõi thông tin lớp.
- Tra cứu hồ sơ học sinh: Xem thông tin cá nhân và kết quả học tập.
- Quản lý điểm số: Nhập và cập nhật kết quả học tập của học sinh.
- Thông báo nội bộ: Gửi thông báo đến lớp học hoặc học sinh.
- Quản lý thời khóa biểu: Cập nhật và theo dõi lịch học, lịch thi của lớp phụ trách.

2.1.4. Chức năng dành cho học sinh (Student)

- Xem thông tin cá nhân: Tra cứu hồ sơ cá nhân và thông tin liên quan đến lớp học.
- Xem điểm số: Theo dõi kết quả học tập theo môn học và học kỳ.
- Xem thời khóa biểu và nhận thông báo: Truy cập lịch học, lịch thi và nhận thông báo từ giáo viên hoặc nhà trường.

2.2. Sơ đồ Use Case

2.2.1. Các tác nhân (Actor)

Trong hệ thống cổng thông tin trường THPT, các tác nhân (actor) chính được xác định dựa trên vai trò và chức năng tương tác với hệ thống. Việc phân loại tác nhân giúp xác định rõ phạm vi hoạt động, quyền hạn và các chức năng tương ứng trong sơ đồ Use Case.

Admin (Quản trị viên)

Là người quản trị hệ thống, thường là cán bộ kỹ thuật hoặc ban giám hiệu. Có toàn quyền trong việc cấu hình hệ thống và quản lý dữ liệu.

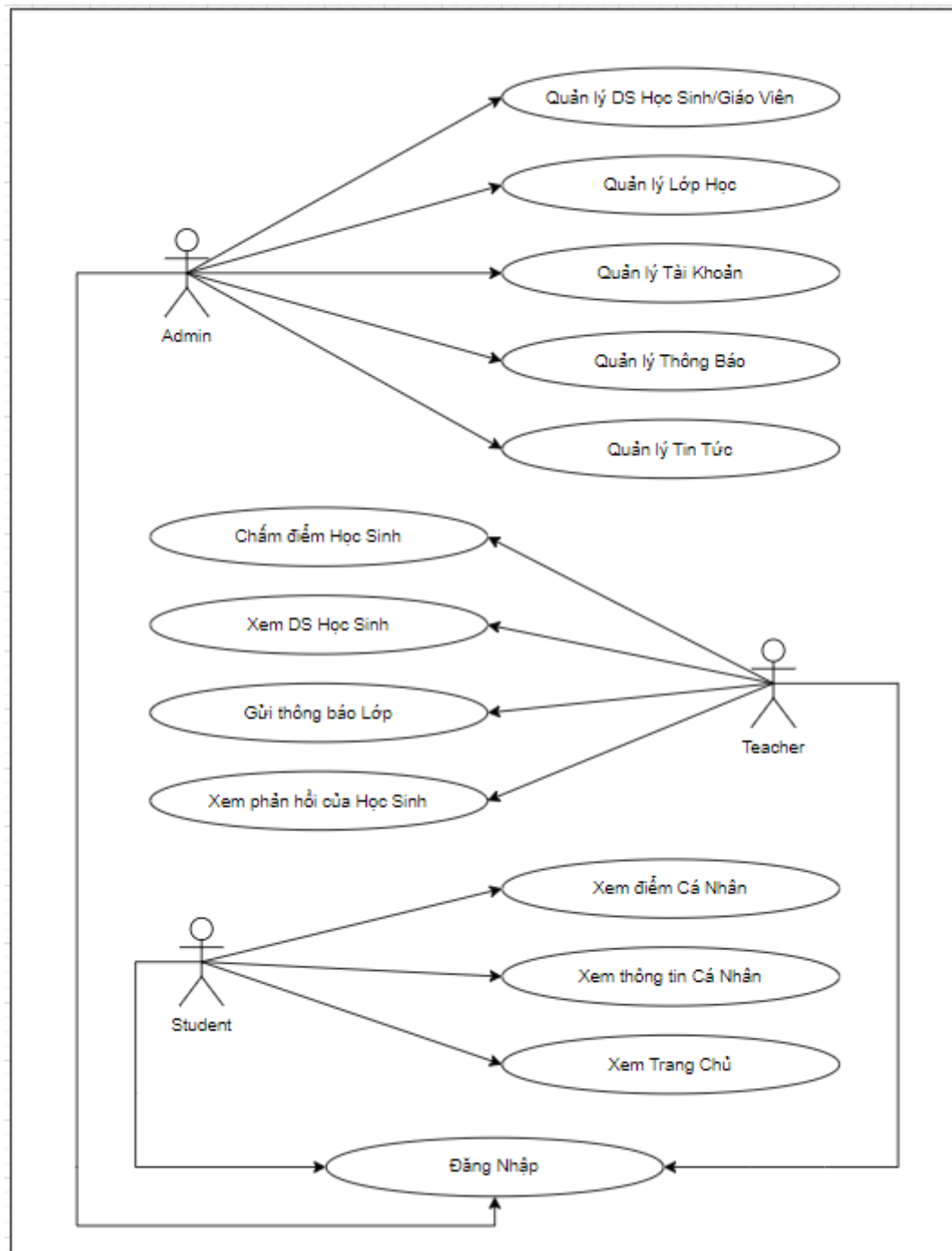
Teacher (Giáo viên)

Là giáo viên giảng dạy tại trường, có quyền truy cập vào các chức năng liên quan đến lớp học và học sinh.

Student (Học sinh)

Là học sinh đang theo học tại trường, có quyền truy cập vào các chức năng tra cứu thông tin cá nhân và học tập.

2.2.2. Biểu đồ Use Case tổng quát



Hình 2.1 Use Case Tổng Quát

2.2.3. Mô tả các Use Case chính

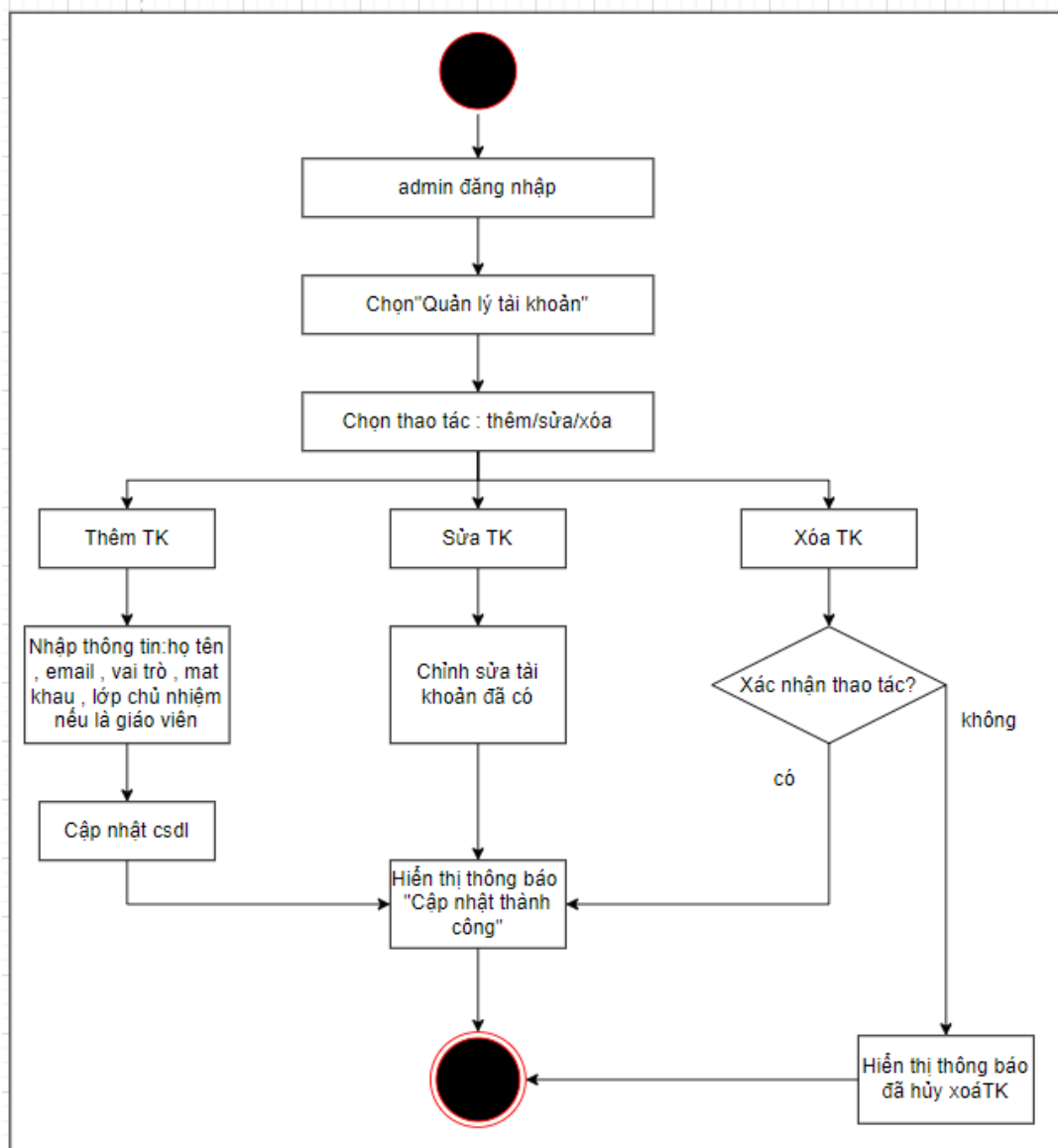
Bảng 2.1 mô tả các use case

Use Case	AcTor	Mô Tả	Tiền Điều Kiện	Hậu Điều Kiện
Quản lý tài khoản	Admin	Admin có thể thêm, sửa hoặc xóa tài khoản của giáo viên và học sinh trong hệ thống.	Admin đã đăng nhập vào hệ thống.	Cơ sở dữ liệu người dùng được cập nhật.
Quản lý lớp học	Admin	Admin tạo, chỉnh sửa hoặc xóa thông tin lớp học; gán giáo viên chủ nhiệm cho lớp.	Admin đã đăng nhập vào hệ thống.	Thông tin lớp học được cập nhật trong CSDL.
Đăng tin tức	Admin	Admin đăng, chỉnh sửa hoặc xóa các bài viết, thông báo của trường hiển thị trên trang chủ.	Admin đã đăng nhập vào hệ thống.	Tin tức được lưu và hiển thị công khai trên website.
Thống kê	Admin	Admin tổng hợp dữ liệu học sinh, điểm số, lớp học để phục vụ quản lý.	Admin đã đăng nhập vào hệ thống.	Các báo cáo thống kê được tạo ra và lưu trữ phục vụ đánh giá, quản lý.
Nhập điểm học sinh	Teacher	Giáo viên nhập hoặc chỉnh sửa điểm của học sinh theo từng môn học và học kỳ.	Giáo viên đã đăng nhập và được phân công lớp phụ trách.	Điểm học sinh được cập nhật vào hệ thống.
Gửi thông báo lớp	Teacher	Giáo viên gửi thông báo đến toàn bộ học sinh trong lớp của mình (ví dụ: lịch học, kiểm tra).	Giáo viên đã đăng nhập.	Học sinh trong lớp nhận được thông báo mới.

Xem điểm học tập	Student	Học sinh xem điểm học tập của bản thân theo từng môn và học kỳ.	Học sinh đã đăng nhập vào hệ thống.	Bảng điểm hiển thị đầy đủ và chính xác.
Xem thông báo	Student	Học sinh xem các thông báo do giáo viên hoặc nhà trường gửi.	Học sinh đã đăng nhập.	Học sinh nắm được thông tin mới từ lớp hoặc trường.
Xem tin tức trường	Student	Học sinh xem các tin tức, hoạt động hoặc thông báo chung của nhà trường.	Học sinh đã đăng nhập.	Tin tức hiển thị đúng theo thời gian đăng tải.

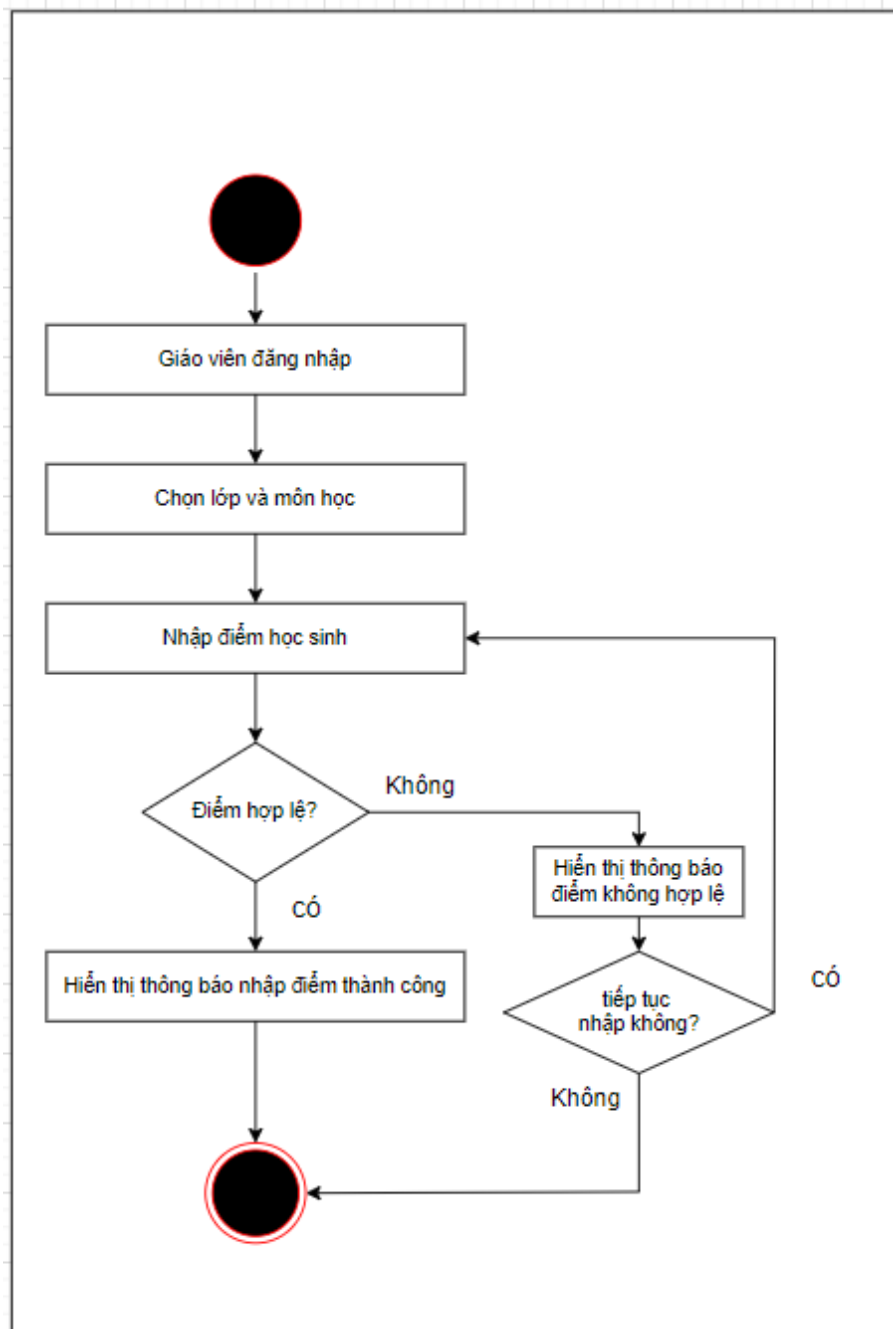
2.3. Sơ đồ hoạt động

2.3.1. Admin - Quản lý tài khoản



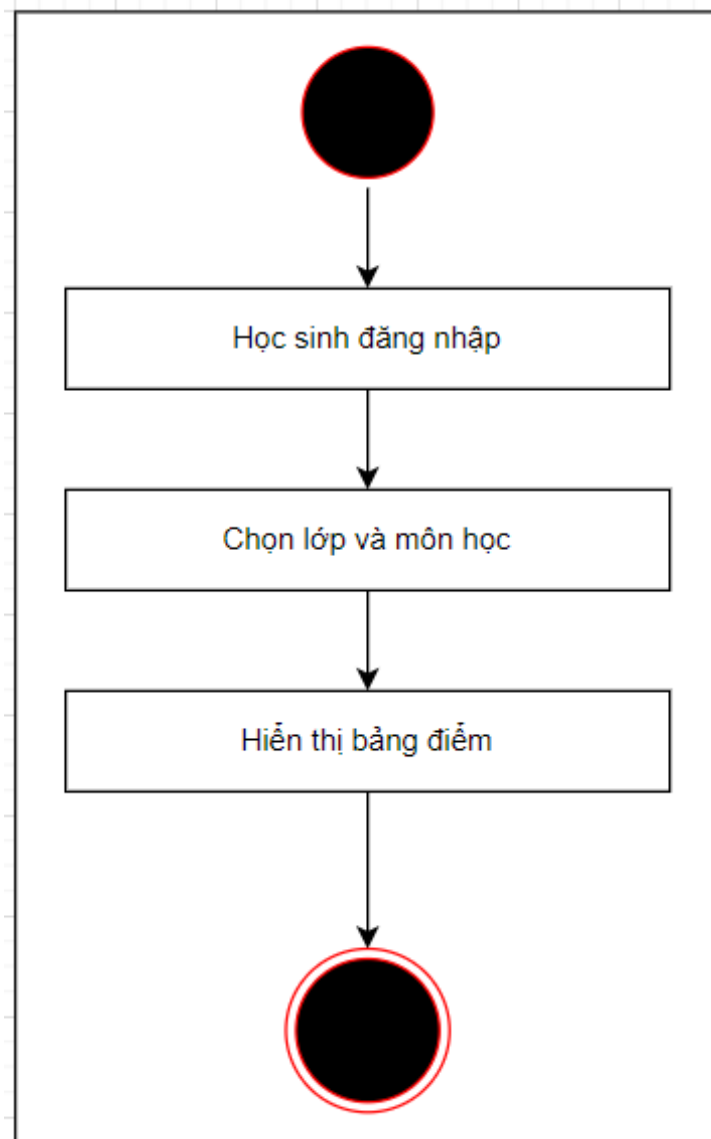
Hình 2.2 Sơ đồ hoạt động – Admin

2.3.2. Teacher - Nhập điểm học sinh



Hình 2.3 Sơ đồ hoạt động– Teacher

2.3.3. Student - Xem điểm học tập



Hình 2.4 Sơ đồ hoạt động – Student

2.4. Cơ sở dữ liệu

2.4.1. Các bảng dữ liệu chính

Hệ thống sử dụng các bảng dữ liệu sau để lưu trữ thông tin và đảm bảo hoạt động của các chức năng:

Bảng 2.2 Các bảng dữ liệu chính

Tên bảng	Mô tả chức năng chính
Admin	Lưu thông tin tài khoản người dùng (Admin, Giáo viên, Học sinh)
HọcSinh	Lưu thông tin cá nhân của học sinh (họ tên, ngày sinh, địa chỉ, lớp...)
GiaoVien	Lưu thông tin giáo viên (họ tên, bộ môn, lớp phụ trách...)
LopHoc	Lưu thông tin lớp học (tên lớp, khối, danh sách học sinh)
MonHoc	Lưu danh sách môn học trong hệ thống
BangDiem	Lưu điểm số của học sinh theo từng môn và học kỳ
PhanCong	Phân công giáo viên dạy môn học cho lớp
Lich	Lịch của từng lớp theo môn, tiết, thứ/hay lịch dạy của giáo viên đó
ThongBao	Lưu thông báo nội bộ gửi đến lớp (do giáo viên tạo)/Lưu thông báo phổ biến đến cả trường (do Admin tạo).
TinTuc	Lưu tin tức công khai do Admin đăng tải
LogHoatDong	Ghi lại các thao tác thêm/sửa/xóa của người dùng để phục vụ kiểm tra và bảo mật hệ thống

2.4.2. Mỗi quan hệ giữa các bảng

Trong hệ thống quản lý thông tin trường THPT, các bảng dữ liệu được thiết kế theo mô hình quan hệ, đảm bảo tính logic, dễ truy vấn và mở rộng. Dưới đây là mô tả chi tiết các mối quan hệ giữa các bảng chính:

Admin – TinTuc

- Một quản trị (Admin) có thể đăng nhiều tin tức.
- Quan hệ 1–N

Admin – ThongBao

- Một quản trị (Admin) có thể gửi nhiều thông báo cho các lớp.
- Quan hệ 1–N

Admin – LogHoatDong

- Mỗi thao tác thêm/sửa/xóa đều được ghi lại theo người thực hiện.
- Quan hệ 1–N

GiaoVien – PhanCong

- Một giáo viên có thể được phân công nhiều lớp/môn.
- Quan hệ 1–N

HocSinh – LopHoc

- Một lớp có nhiều học sinh, mỗi học sinh thuộc một lớp.
- Quan hệ 1–N

HocSinh – BangDiem

- Một học sinh có nhiều điểm theo từng môn/học kỳ.
- Quan hệ 1–N

MonHoc – BangDiem

- Một môn học có nhiều bản ghi điểm (vì nhiều học sinh cùng học môn đó).
- Quan hệ 1–N

LopHoc – PhanCong

- Một lớp có thể có nhiều phân công (nhiều giáo viên cho nhiều môn)
- Quan hệ 1–N.

MonHoc – PhanCong

- Một môn học có thể xuất hiện trong nhiều phân công (ở các lớp khác nhau)
- Quan hệ 1–N.

LopHoc – Lich

- Mỗi lớp có nhiều lịch học
- Quan hệ 1–N

MonHoc – Lich

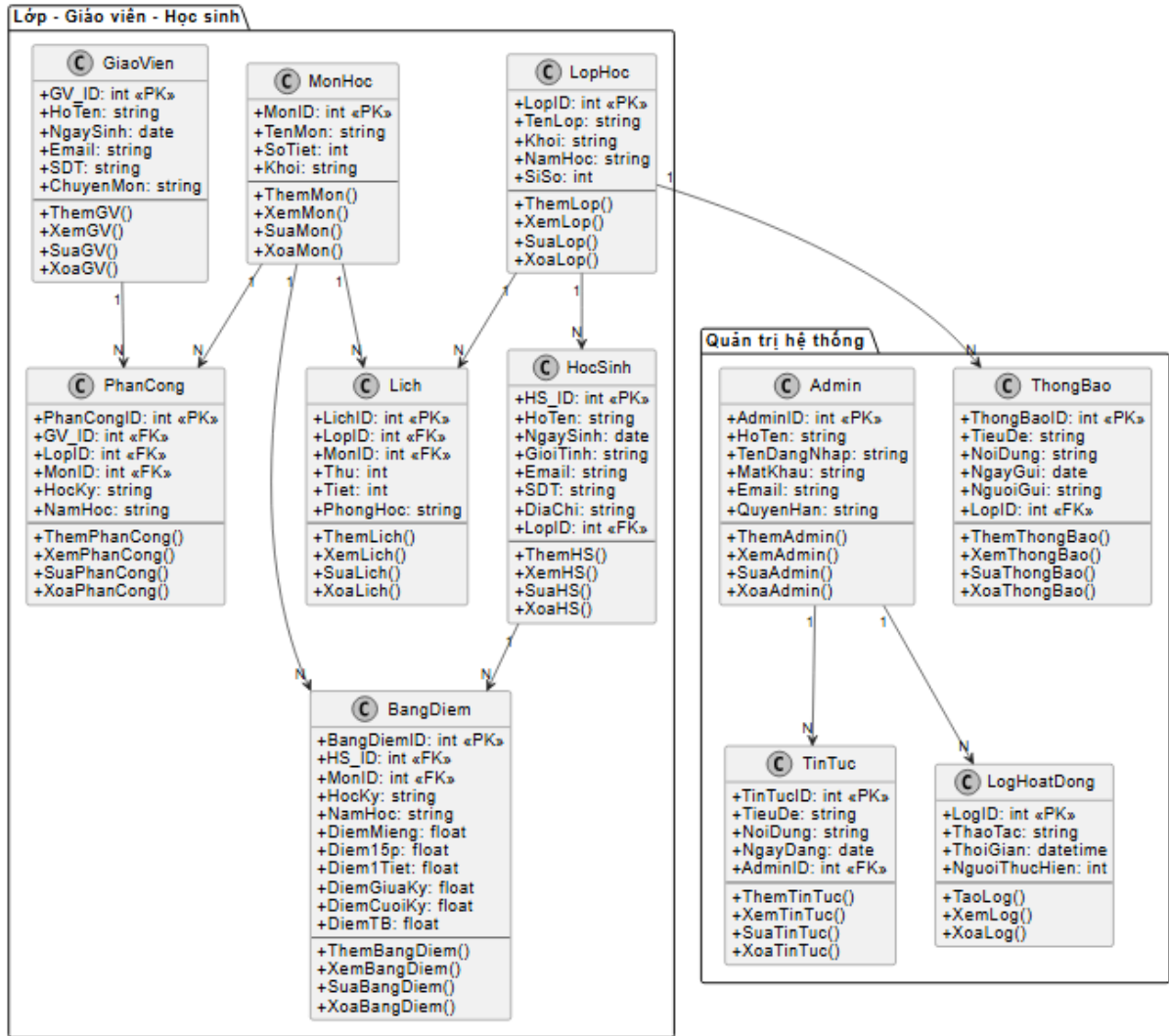
- Mỗi môn học được chia theo nhiều lịch của lớp
- Quan hệ 1–N

LopHoc – ThongBao

- Một lớp có thể nhận nhiều thông báo khác nhau (từ admin hay từ giáo viên).
- Quan hệ 1–N

2.4.3. Mô hình ERD

Mô hình ERD (Entity Relationship Diagram) là sơ đồ biểu diễn các thực thể (entity) trong hệ thống và mối quan hệ giữa chúng. Trong hệ thống cổng thông tin trường THPT, các thực thể chính được xác định dựa trên chức năng quản lý thông tin người dùng, lớp học, môn học, điểm số, thời khóa biểu và thông báo.



Hình 2.5 Sơ đồ erd

CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

3.1. Giao diện học sinh

Chức năng xem thông tin cá nhân

 Thông tin cá nhân

Thông tin học sinh

Họ tên:	Phạm minh hiếu
Giới tính:	Chưa cập nhật
Email:	student@example.com
Lớp:	10A1(S)
Mã lớp (ID):	16
Mã học sinh:	01
Vai trò:	student
Ngày sinh:	26/06/2003
Số CCCD:	123456789
Số điện thoại:	0328185636
Họ tên cha:	Phạm Minh X
Họ tên mẹ:	Trần Ngọc Y
Số điện thoại phụ huynh:	

Hình 3.1 : Chức năng xem thông tin cá nhân(Student)

Chức năng xem thông tin cá nhân trong giao diện học sinh cho phép mỗi học sinh sau khi đăng nhập vào hệ thống có thể truy cập mục “Thông tin cá nhân” để theo dõi đầy đủ dữ liệu của mình, bao gồm họ tên, ngày sinh, mã số học sinh, lớp, khóa học, thông tin liên hệ và ảnh đại diện; hệ thống hiển thị chi tiết nhưng chỉ cho phép chỉnh sửa một số trường như số điện thoại hoặc email, còn các trường quan trọng (họ tên, mã số, lớp) chỉ được quản lý bởi giáo viên hoặc admin, đảm bảo tính chính xác và bảo mật dữ liệu.

Chức năng xem điểm

Trường THPT Kani
 Nơi chấp cánh ước mơ

[Trang Chủ](#)
[Thông tin cá nhân](#)
[Lịch học](#)
[Điểm](#)
[Nhóm chat](#)
[Đề xuất](#)

Phạm minh hiếu ▾

Bảng điểm của Phạm minh hiếu

STT	Môn học	Điểm miệng	Thái độ	Giữa kỳ	Cuối kỳ	Điểm tổng kết
1	văn	9	9	9	10	9.4
2	GDCD	5	7	3	6	5.1
3	Hóa	9	9	9	9	9

Hình 3.2 : Chức năng xem điểm(Student)

Chức năng xem điểm trong giao diện học sinh cho phép mỗi học sinh sau khi đăng nhập vào hệ thống có thể truy cập mục “Kết quả học tập” để theo dõi toàn bộ điểm số của mình theo từng môn học và từng học kỳ; hệ thống hiển thị chi tiết các loại điểm như điểm miệng, điểm kiểm tra thường xuyên, điểm giữa kỳ, điểm cuối kỳ và điểm trung bình, đồng thời sắp xếp theo môn học và thời gian để học sinh dễ dàng nắm bắt tình hình học tập, đảm bảo tính minh bạch và chính xác trong quá trình quản lý kết quả học tập.

3.2. Giao diện quản lý

Chức năng quản lý danh sách học sinh


[Trang Chủ](#)
[Quản lý học sinh](#)
[Quản lý điểm](#)
[Quản lý Lớp Học](#)
[Quản lý Lịch Học](#)
[Quản lý Trang chủ](#)
[+ Xem thêm](#)

Admin Master ▾

Quản Lý Học Sinh

Cấp phát 1 học sinh

Cấp phát nhiều học sinh

STT	Họ và tên	Mã học sinh	Lớp	Email	Thao tác
1	Phạm minh hiếu	01	10A1(S)	student@example.com	Xem Sửa Xóa
2	Admin User	02	10A1(S)	ctlt.cancer@gmail.com	Xem Sửa Xóa
3	kaniheikagani	2152132346	10A1(C)	hiou2152132346@gmail.com	Xem Sửa Xóa
4	Đặng Minh Khôi	10A004	10A1(S)	khoi.dang@example.com	Xem Sửa Xóa
5	Đặng Thảo Vy	10A005	10A1(S)	vy.dang@example.com	Xem Sửa Xóa
6	Lưu Đức Anh	10A006	10A1(S)	duc.anh@example.com	Xem Sửa Xóa

Hình 3.3 : Chức năng quản lý danh sách học sinh(Admin)

Chức năng quản lý danh sách học sinh trong giao diện quản lý cho phép Admin hoặc giáo viên chủ nhiệm theo dõi và thao tác trực tiếp trên dữ liệu học sinh của từng lớp. Khi truy cập mục này, hệ thống hiển thị đầy đủ danh sách học sinh kèm theo các thông tin cơ bản như họ tên, mã số học sinh, ngày sinh, giới tính, lớp, và thông tin liên hệ. Người quản lý có thể thực hiện các thao tác như thêm mới học sinh, chỉnh sửa thông tin cá nhân, xóa học sinh khỏi danh sách, hoặc tìm kiếm và lọc theo tiêu chí (ví dụ: theo lớp, theo tên). Chức năng này giúp đảm bảo dữ liệu học sinh luôn được cập nhật chính xác, hỗ trợ việc quản lý lớp học hiệu quả và thuận tiện.

Chức năng phân công giáo viên


[Trang Chủ](#)
[Quản lý học sinh](#)
[Quản lý điểm](#)
[Quản lý Lớp Học](#)
[Quản lý Lịch Học](#)
[Quản lý Trang chủ](#)
[+ Xem thêm](#)

Admin Master

Danh sách lớp học

Thêm lớp mới

STT	Tên lớp	Giáo viên phụ trách	Giáo viên bộ môn	Hành động
1	10A1(S)	Thầy Nguyễn Văn A	Toán: Thầy Nguyễn Văn A Văn: Cô Trần Thị B Lý: Thầy Lê Văn C Anh: Cô Phạm Thị D Địa: Thầy Nguyễn Văn E Hóa: Cô Trần Thị F Sinh: Thầy Lê Văn G GDCD: Cô Phạm Thị H Sử: Thầy Nguyễn Văn I Tin: Cô Trần Thị K Lý: Thầy Nguyễn Văn E Lý: Thầy Nguyễn Văn E Lý: Thầy Nguyễn Văn E Toán: Thầy Nguyễn Văn A	Bộ môn Sửa Xóa
2	10A1(C)	Cô Trần Thị B	Toán: Thầy Nguyễn Văn A Văn: Cô Trần Thị B Lý: Thầy Lê Văn C Anh: Cô Phạm Thị D	Bộ môn Sửa Xóa

Hình 3.4 : Chức năng phân công giáo viên(Admin)

Chức năng phân công giáo viên trong giao diện quản lý cho phép Admin hoặc giáo viên chủ nhiệm thực hiện việc gán giáo viên vào các lớp và môn học cụ thể. Khi truy cập mục này, hệ thống hiển thị danh sách lớp học và môn học hiện có, đồng thời cung cấp danh sách giáo viên để lựa chọn. Người quản lý có thể tạo mới một bản ghi phân công bằng cách chọn lớp, chọn môn học và chọn giáo viên phụ trách, sau đó lưu lại để hệ thống cập nhật. Ngoài ra, chức năng còn hỗ trợ chỉnh sửa hoặc xóa phân công khi có sự thay đổi về giáo viên hoặc môn học. Nhờ đó, việc quản lý giảng dạy trở nên minh bạch, dễ kiểm soát và đảm bảo mỗi lớp, mỗi môn đều có giáo viên phụ trách rõ ràng.

3.3. Giao diện giáo viên

Chức năng nhập điểm(Teacher)


[Trang Chủ](#)
[Danh sách học sinh](#)
[Điểm lớp chủ nhiệm](#)
[Nhập điểm lớp phân công](#)
[Quản lý nhóm lớp](#)
[Lịch dạy](#)
[+ Xem thêm](#)

Thầy Nguyễn Văn A ▾

Nhập điểm lớp 10A1(S)


 Chọn lớp bạn được phân công:

Lớp 10A1(S) ▾

Xem danh sách

STT	Họ và tên	Môn học	Điểm miệng	Thái độ	Giữa kỳ	Cuối kỳ	Điểm tổng kết	Thao tác
1	Phạm minh hiếu	Toán ▾						Lưu
2	Admin User	Toán ▾						Lưu
3	Đặng Minh Khôi	Toán ▾						Lưu
4	Đặng Thảo Vy	Toán ▾						Lưu
5	Lưu Đức Anh	Toán ▾						Lưu

Hình 3.5 : Chức năng nhập điểm(Teacher)

Chức năng nhập điểm trong giao diện giáo viên cho phép mỗi giáo viên sau khi đăng nhập vào hệ thống có thể truy cập mục “Nhập điểm” để ghi nhận kết quả học tập của học sinh theo từng môn học và từng lớp mà mình phụ trách. Hệ thống hiển thị danh sách học sinh trong lớp cùng với các trường nhập liệu cho các loại điểm như điểm miệng, điểm kiểm tra thường xuyên, điểm giữa kỳ, điểm cuối kỳ và điểm trung bình. Giáo viên nhập dữ liệu trực tiếp, sau đó lưu lại để hệ thống cập nhật vào bảng điểm chung. Chức năng này đảm bảo việc quản lý kết quả học tập được thực hiện minh bạch, chính xác và thuận tiện, đồng thời hỗ trợ việc theo dõi tiến độ học tập của từng học sinh.

Chức năng xem danh sách học sinh(Teacher)

 Trang Chủ	Danh sách học sinh	Điểm lớp chủ nhiệm	Nhập điểm lớp phân công	Quản lý nhóm lớp	Lịch dạy	+ Xem thêm	Thầy Nguyễn Văn A ▾
---	------------------------------------	------------------------------------	---	----------------------------------	--------------------------	----------------------------	---------------------

 Học sinh lớp 10A1(S)
--

STT	Họ và tên	Email	Ngày Sinh	sdt
1	Phạm minh hiếu	student@example.com	26/06/2003	0328185636
2	Admin User	ctlt.cancer@gmail.com	Chưa cập nhật	Chưa cập nhật
3	Đặng Minh Khôi	khoi.dang@example.com	Chưa cập nhật	Chưa cập nhật
4	Đặng Thảo Vy	vy.dang@example.com	Chưa cập nhật	Chưa cập nhật
5	Lưu Đức Anh	duc.anh@example.com	Chưa cập nhật	Chưa cập nhật

Hình 3.6 : Chức năng xem danh sách học sinh(teacher)

Chức năng xem danh sách học sinh trong giao diện giáo viên cho phép mỗi giáo viên sau khi đăng nhập vào hệ thống có thể truy cập mục “Danh sách học sinh” để theo dõi toàn bộ học sinh trong lớp mà mình phụ trách. Hệ thống hiển thị đầy đủ thông tin cơ bản của từng học sinh như họ tên, mã số học sinh, ngày sinh, giới tính, lớp và thông tin liên hệ. Giáo viên có thể sử dụng chức năng tìm kiếm hoặc lọc theo tiêu chí (ví dụ: theo tên, theo mã số) để nhanh chóng tra cứu. Chức năng này giúp giáo viên quản lý lớp học thuận tiện hơn, hỗ trợ việc theo dõi tình hình học tập và liên hệ với học sinh khi cần thiết.

KẾT LUẬN

Qua quá trình nghiên cứu và thực hiện đề tài, nhóm đã đạt được những kết quả đáng ghi nhận như sau:

- Hệ thống được xây dựng đáp ứng đầy đủ các chức năng theo yêu cầu đề ra, đảm bảo tính ổn định và hiệu quả trong quá trình vận hành.
- Giao diện người dùng được thiết kế thân thiện, dễ sử dụng, phù hợp với đối tượng mục tiêu.
- Việc xử lý dữ liệu và bảo mật thông tin được chú trọng, đảm bảo an toàn trong quá trình truy cập và sử dụng hệ thống.
- Đề tài đã góp phần làm rõ quy trình phát triển một ứng dụng web từ khâu phân tích đến triển khai thực tế.

Đề xuất và kiến nghị:

- Tiếp tục mở rộng chức năng hệ thống để phục vụ nhiều đối tượng hơn.
- Tích hợp thêm các công nghệ mới như AI hoặc chatbot để nâng cao trải nghiệm người dùng.
- Cần có sự phối hợp chặt chẽ giữa các bộ phận kỹ thuật và nghiệp vụ để đảm bảo tính thực tiễn khi triển khai.

xây dựng website Cổng thông tin của trường THPT

TÀI LIỆU THAM KHẢO

PHỤ LỤC 1

PHỤ LỤC 2