KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI MÃ NGUỒN MỞ**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ SINH VIÊN**

Giảng viên giảng dạy : **Nguyễn Ngọc Đan Thanh**

Sinh viên thực hiện:

Lớp: **DA21TTC**

**Nguyễn Hoàng Thương 110121224**

**Lê Thị Ngọc Hân 110121249**

**Ngô Thanh Quyền 110121223**

**Nguyễn Khánh Băng 110121248**

**Trà Vinh, tháng 10 năm 2024**

**LỜI CẢM ƠN**

**🖎🕮✍**

Nhóm em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các Thầy/Cô của Trường Đại học Trà Vinh, đặc biệt là Cô Nguyễn Ngọc Đan Thanh đã tận tình truyền đạt kiến thức trong môn phát triển ứng dụng web với mã nguồn mở. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu bài báo cáo mà nó còn là hành trang quý báu để nhóm em áp dụng vào thực tế một cách vững chắc và tự tin.

Chúng em cũng xin chân thành cám ơn Cô Nguyễn Ngọc Đan Thanh là người hướng dẫn giúp đỡ và cung cấp những kiến thức quý báu giúp chúng em hoàn thành tốt bài báo cáo của nhóm mình.

Do còn hạn chế về kiến thức cũng như những kinh nghiệm thực tế cho nên không tránh khỏi được những sai sót trong quá trình tìm hiểu và trình bày rất mong nhận được sự đóng góp ý kiến của thầy để bài báo cáo được hoàn chỉnh hơn.

Sau cùng, nhóm em xin kính chúc Cô Nguyễn Ngọc Đan Thanh thật nhiều sức khoẻ, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Xin chân thành cảm ơn!

**MỤC LỤC**

Nội dung Trang

[CHƯƠNG 1: TỔNG QUAN 5](#_Toc181028407)

[1.1 Mô tả bài toán 7](#_Toc181028408)

[1.2 Công nghệ sử dụng 8](#_Toc181028409)

[1.2.1 PHP 8](#_Toc181028410)

[1.2.2 MySQL 8](#_Toc181028411)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 10](#_Toc181028412)

[2.1 Phân trang 10](#_Toc181028413)

[2.1.1 Phân trang là gì? 10](#_Toc181028414)

[2.1.2 Các bước phân trang trong php 10](#_Toc181028415)

[2.2 Đăng nhập bằng tài khoản google 12](#_Toc181028416)

[2.3 Công nghệ AJAX trong PHP 16](#_Toc181028417)

[2.3.1 Khái niệm AJAX trong PHP 16](#_Toc181028418)

[2.3.2 Cách hoạt động 16](#_Toc181028419)

[2.3.3 Ứng dụng của AJAX 17](#_Toc181028420)

[2.4 Rewirite 22](#_Toc181028421)

[2.5 Ứng dụng Javascript gọi API hiển thị địa lý 24](#_Toc181028422)

[CHƯƠNG 3: THỰC HIỆN HÓA NGHIÊN CỨU 28](#_Toc181028423)

[3.1 Thiết kế cơ sở dữ liệu 28](#_Toc181028424)

[3.1.1 Cấu trúc cơ sở dữ liệu 28](#_Toc181028425)

[3.1.2 Mô hình ràng buộc cơ sở dữ liệu 28](#_Toc181028426)

[3.2 Kết nối cơ sở dữ liệu 30](#_Toc181028427)

[3.2.1 Tạo CSDL 30](#_Toc181028428)

[3.2.2 Kết nối cơ sở dữ liệu 31](#_Toc181028429)

[CHƯƠNG 4: XÂY DỰNG HỆ THỐNG 32](#_Toc181028430)

[4.1 Đăng ký 32](#_Toc181028431)

[4.2 Đăng nhập 33](#_Toc181028432)

[4.3 Phân quyền 34](#_Toc181028433)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 40](#_Toc181028434)

[5.1 Kết luận 40](#_Toc181028435)

[5.1.1 Kết quả đạt được 40](#_Toc181028436)

[5.1.2 Hạn chế 40](#_Toc181028437)

[5.2 Hướng phát triển 40](#_Toc181028438)

[DANH MỤC TÀI LIỆU THAM KHẢO 41](#_Toc181028439)

**DANH MỤC HÌNH ẢNH**

[Hình 2. 1 Ví dụ về phân trang 10](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028750)

[Hình 2. 2 Create new project trên Google Developer Console. 12](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028751)

[Hình 2. 3 Create Credentials 12](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028752)

[Hình 2. 4 OAuth Client IDs 13](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028753)

[Hình 2. 5 Client ID for Web application 13](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028754)

[Hình 2. 6 Thông tin Client Id, secret 14](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028755)

[Hình 2. 7 Cách hoạt động của AJAX 16](#_Toc181028756)

[Hình 2. 8 Ví dụ ứng dụng AJAX tìm kiếm sinh viên 17](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028757)

[Hình 2. 9 Ví dụ ứng dụng AJAX lọc dữ liệu 20](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028758)

[Hình 2. 10 Ví dụ Rewirite trang login.php 24](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028759)

[Hình 2. 11 Ví dụ lấy địa chỉ từ API 27](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028760)

[Hình 2. 12 Tạo thủ công cơ sở dữ liệu mới 30](#_Toc181028761)

[Hình 2. 13 Nhập tên csdl mới 30](#_Toc181028762)

[Hình 2. 14 Tạo csdl bằng lệnh 31](#_Toc181028763)

[Hình 3. 1 Mô hình ràng buộc cơ sở dữ liệu 29](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181028764)

[Hình 4. 1 Đăng ký 32](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029022)

[Hình 4. 2 Đăng nhập 33](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029023)

[Hình 4. 3 Thông tin người dùng Google đã lưu thành công 34](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029024)

[Hình 4. 4 Quyền admin 34](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029025)

[Hình 4. 5 Thêm lớp học mới 35](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029026)

[Hình 4. 6 Xóa lớp học 35](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029027)

[Hình 4. 7 Sửa thông tin lớp học 36](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029028)

[Hình 4. 8 Thêm sinh viên 37](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029029)

[Hình 4. 9 Xóa sinh viên 38](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029030)

[Hình 4. 10 Sửa thông tin sinh viên 38](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029031)

[Hình 4. 11 Quyền user 39](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029032)

[Hình 4. 12 Xem thông tin sinh viên 39](file:///D:\.download\New%20folder\220243_nhom5\Ly_thuyet\Bao_cao_ly_thuyet_nhom5.docx#_Toc181029033)

**CHƯƠNG 1: TỔNG QUAN**

**1.1 Mô tả bài toán**

Hệ thống quản lý sinh viên bằng PHP và MySQL nhằm mục đích cung cấp một ứng dụng web giúp quản lý thông tin sinh viên và lớp học trong các trường học hoặc trung tâm đào tạo. Hệ thống hỗ trợ việc thêm mới, sửa đổi, xóa và tra cứu thông tin lớp học và sinh viên, đồng thời phân quyền quản lý giữa các vai trò người dùng khác nhau (admin và user).

Hệ thống cần cung cấp các chức năng quản lý thông tin lớp học và sinh viên như sau:

* **Quản lý thông tin lớp học:**

Thêm lớp học: Cho phép admin tạo mới các lớp học với thông tin chi tiết như tên lớp, mã lớp, và mô tả. Hỗ trợ nhập các thông tin cần thiết để phân loại và quản lý các lớp học một cách hiệu quả.

Xóa lớp học: Cho phép admin xóa những lớp học không còn hoạt động. Khi xóa lớp học, hệ thống sẽ tự động xóa tất cả sinh viên thuộc lớp đó để đảm bảo tính nhất quán dữ liệu.

Sửa lớp học: Admin có thể chỉnh sửa thông tin lớp học hiện có như tên lớp, mã lớp, hoặc các thông tin khác.

* **Quản lý thông tin sinh viên**

Thêm sinh viên: Cho phép admin thêm thông tin sinh viên mới vào hệ thống, bao gồm các trường thông tin như tên, mã sinh viên, lớp học, ngày sinh, và ảnh đại diện.

Các điều kiện bắt buộc:

* Mã sinh viên phải là số nguyên duy nhất, không trùng lặp.
* Sinh viên phải đủ 18 tuổi trở lên tính đến ngày đăng ký.
* Ảnh đại diện phải có định dạng JPG, JPEG, PNG, hoặc GIF, và kích thước không vượt quá 500kb.

Xóa sinh viên: Admin có thể xóa thông tin sinh viên khi sinh viên rời trường hoặc tốt nghiệp.

Sửa thông tin sinh viên: Chỉnh sửa thông tin sinh viên hiện có, bao gồm tên, lớp học, ngày sinh, hoặc ảnh.

* **Phân quyền và chức năng người dùng**

Hệ thống chia thành hai vai trò chính:

Admin: Có quyền quản lý tất cả các thông tin trong hệ thống, bao gồm thêm, sửa, xóa lớp học và sinh viên.

User:

* Xem thông tin sinh viên: Cho phép người dùng tra cứu thông tin cá nhân trong hệ thống.
* Chỉnh sửa thông tin cá nhân: Người dùng có thể cập nhật các thông tin cơ bản như ảnh đại diện hoặc thông tin liên hệ.

**1.2 Công nghệ sử dụng**

**1.2.1 PHP**

PHP (PHP Hypertext Preprocessor) là ngôn ngữ Server-side script được thực thi trong WebServer. Ngôn ngữ PHP cung cấp cho người dùng cơ sở hạ tầng chặt chẽ bao gồm giao thức, API, cơ sở dữ liệu,… Ngôn ngữ này được sử dụng để lập trình website.

PHP được tích hợp dễ dàng với HTML, giúp xây dựng các trang web tương tác. Các tập lệnh PHP được thực thi trên máy chủ, sau đó kết quả được trả về trình duyệt dưới dạng HTML.

Ưu điểm của PHP là miễn phí, mã nguồn mở, dễ học và có cộng đồng hỗ trợ lớn. PHP tương thích với nhiều hệ điều hành và máy chủ web khác nhau, như Apache, Nginx, và IIS.

PHP thường được sử dụng để xử lý các tác vụ như: xử lý biểu mẫu, tạo trang web động, quản lý phiên làm việc (sessions), và tương tác với cơ sở dữ liệu.

**1.2.2 MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, sử dụng ngôn ngữ SQL (Structured Query Language) để quản lý và truy xuất dữ liệu.

MySQL lưu trữ thông tin dưới dạng các bảng dữ liệu có liên quan đến nhau. Mỗi bảng bao gồm các dòng (bản ghi) và cột (trường).

Được sử dụng rộng rãi cho các ứng dụng web nhờ vào tính hiệu quả, tốc độ và độ tin cậy. MySQL thường được dùng kết hợp với PHP để xây dựng các hệ thống quản lý nội dung, diễn đàn, cửa hàng trực tuyến, và nhiều ứng dụng web khác.

Một số thao tác cơ bản trong MySQL bao gồm: thêm mới, cập nhật, xóa dữ liệu , truy vấn dữ liệu bằng các câu lệnh SQL, và thiết lập các mối quan hệ giữa các bảng dữ liệu.

Đối với hệ thống quản lý sinh viên MySQL được sử dụng làm cơ sở dữ liệu để lưu trữ thông tin về lớp học, sinh viên, và người dùng.

HTML/CSS/JavaScript: Tạo giao diện người dùng và tăng cường trải nghiệm người dùng với các tính năng tương tác như AJAX.

* **Các công nghệ liên quan khác:**

HTML/CSS/JavaScript: Được sử dụng để xây dựng giao diện người dùng (UI).

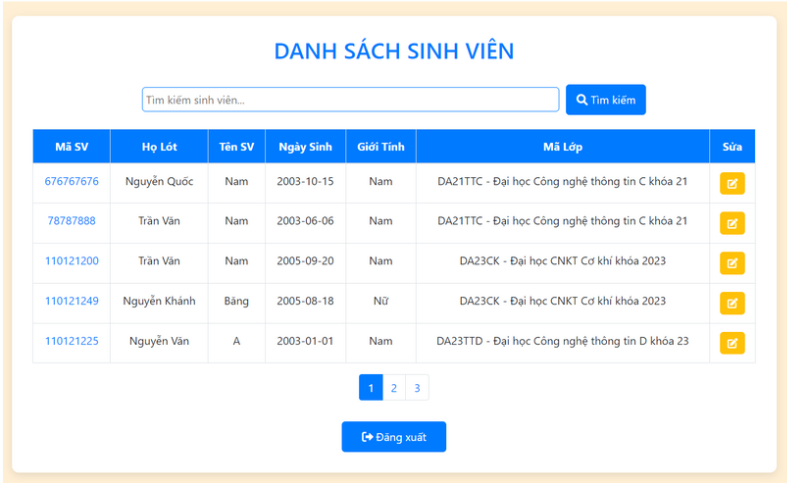
* HTML: Tạo cấu trúc cho trang web.
* CSS: Định dạng giao diện và trình bày nội dung.
* JavaScript: Tăng cường khả năng tương tác và động cho trang web.

AJAX (Asynchronous JavaScript and XML): Kỹ thuật kết hợp giữa JavaScript và các lệnh gọi tới máy chủ để cập nhật nội dung của trang web mà không cần tải lại toàn bộ trang. Điều này giúp tăng tốc trải nghiệm người dùng và làm cho các ứng dụng web trở nên mượt mà hơn.

**CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT**

**2.1 Phân trang**

**2.1.1 Phân trang là gì?**

Phân trang cơ bản là quá trình lấy một tập hợp các kết quả và phân chia thành các trang để dễ xem hơn.Với những trang Web có dữ liệu lớn, việc phân trang là rất quan trọng, nó giúp trang web sẽ load nhanh hơn , giúp người dùng dễ dàng nhấn chọn trang mà mình mong muốn, đến được trang có nội dung mà mình cần, rất nhanh chóng, dễ dàng và chính xác.

Hình 2. Ví dụ về phân trang

**2.1.2 Các bước phân trang trong php**

* **Bước 1: Kết nối csdl**
* **Bước 2: Thiết lập số lượng sinh viên trên mỗi trang và xác định trang hiện tại**

|  |
| --- |
| $limit = 5; // Số sinh viên trên mỗi trang $page = isset($\_GET['page']) && is\_numeric($\_GET['page']) ? (int) $\_GET['page']: 1; |

Đoạn mã trên dùng để thiết lập thông số (limit) cho việc phân trang, xác định số lượng record trên mỗi trang và xác định trang hiện tại dựa trên tham số trong URL, kiểm tra xem có tham số page trong URL không, nếu không, nó mặc định trang hiện tại là 1.

* **Bước 3: Tính toán chỉ số bắt đầu (offset)**

$offset = ($page - 1) \* $limit;

Đoạn mã trên sử dụng để tính toán số lượng trang cần thiết cho phân trang và xác định vị trí bắt đầu (offset) cho dữ liệu mà bạn sẽ lấy từ cơ sở dữ liệu.

* **Bước 4: Tìm tổng số sinh viên và tổng số trang**

|  |
| --- |
| // Đếm tổng số sinh viên  $total\_students = $result->num\_rows;  // Tính tổng số trang  $total\_pages = ceil(num: $total\_students / $limit); |

***num\_rows*** là thuộc tính của đối tượng ***$result*** dùng để lấy số lượng hàng (dòng) trong kết quả truy vấn. Điều này có nghĩa là ***$total\_students*** sẽ chứa tổng số sinh viên được truy vấn từ cơ sở dữ liệu. Sau đó sẽ tính tổng số trang bằng cách chia tổng số sinh viên ***($total\_students***) cho số lượng sinh viên trên mỗi trang ***($limit)*** để xác định số trang cần thiết

* **Bước 5: Truy vấn lấy danh sách sinh viên**

|  |
| --- |
| $sql.= " LIMIT $limit OFFSET $offset"; $result = $conn->query($sql); |

Thêm điều kiện ***LIMIT*** và ***OFFSET*** vào câu lệnh SQL để xác định số lượng record và vị trí bắt đầu của record cho truy vấn.

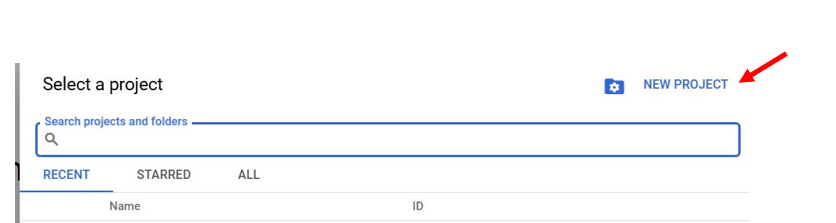
* **Bước 6: Hiển thị danh sách sinh viên và phân trang**

|  |
| --- |
| for ($i = 1; $i <= $total\_pages; $i++) {  $active = $i == $page? 'active':";  echo "<li class='page-item $active'><a class='page-link' href='?  page=$i&search=$search'>$i</a></li>";  } |

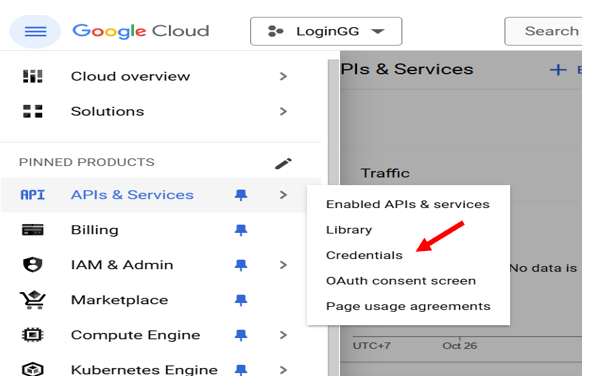
Tạo các liên kết phân trang trong một giao diện người dùng (UI) cho trang web. Đây là cách để người dùng có thể điều hướng qua các trang dữ liệu. Sử dụng vòng lặp for để lặp qua tất cả các số trang từ 1 đến tổng số trang ($total\_pages). Giá trị '**active**' có thể được sử dụng trong CSS để làm nổi bật trang hiện tại trong giao diện, giúp người dùng dễ nhận biết. Tạo một liên kết (<a>) với lớp CSS page-link cho phép người dùng chuyển đến trang $i.

**2.2 Đăng nhập bằng tài khoản google**

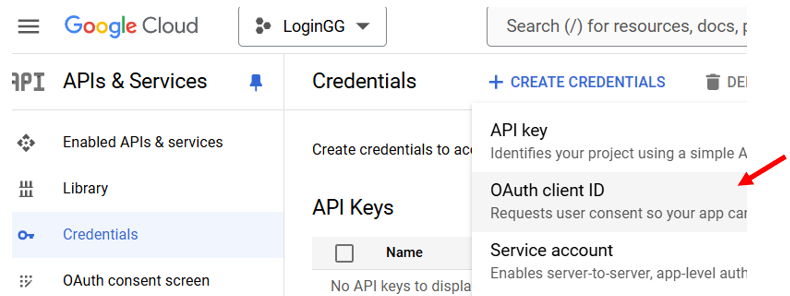
* **Bước 1: Tạo Dự Án Trên Google Developer Console**

1. Truy cập Google Developer Console:
2. Tạo một dự án mới:

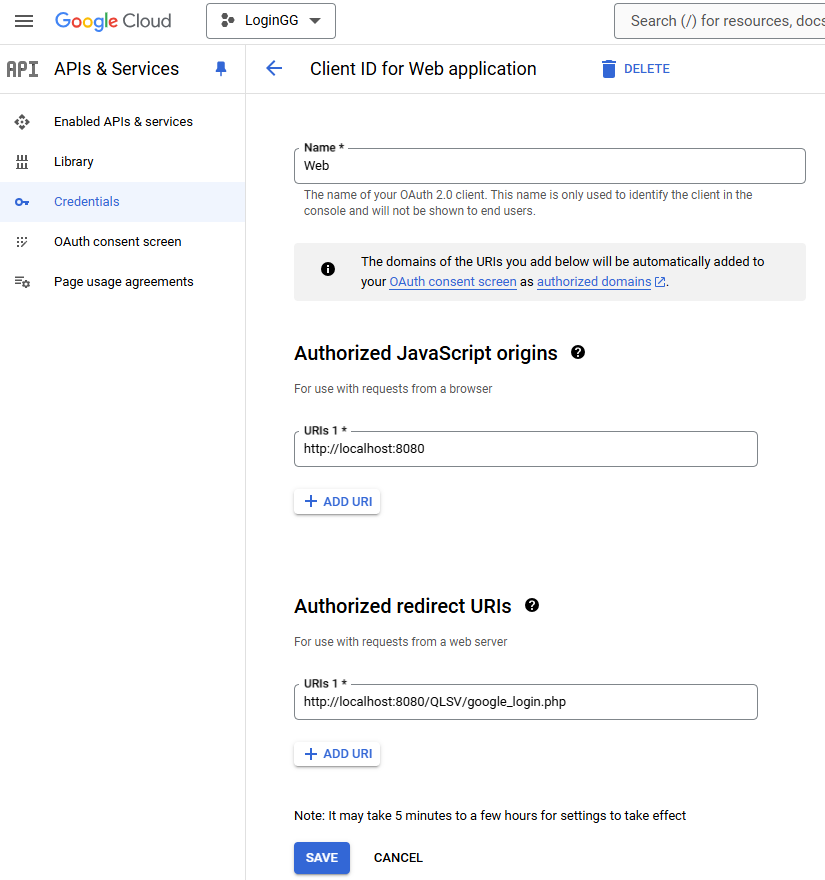
Hình 2. Create new project trên Google Developer Console.

* **Bước 2: Bật API và Tạo Thông Tin Xác Thực**
* Trong bảng điều khiển dự án, sẽ thấy một menu bên trái. Từ đây, đi đến phần “Library” (Thư viện). Tại đây, cần tìm kiếm và bật Google People API hoặc Google+ API. Việc bật API này là cần thiết để ứng dụng của bạn có thể truy cập vào dữ liệu người dùng.
* Tạo thông tin xác thực: Sau khi bật API, quay lại menu bên trái và chọn “Credentials” (Thông tin xác thực).

Hình 2. Create Credentials

* Nhấn vào “Create Credentials” (Tạo thông tin xác thực) và chọn “OAuth Client IDs”. Đây là bước quan trọng để tạo thông tin xác thực cho ứng dụng của bạn.

Hình 2. OAuth Client IDs

* Chọn “Web application” (Ứng dụng web) và điền các thông tin cần thiết như tên ứng dụng và URL chuyển hướng.

Hình 2. Client ID for Web application

* **Bước 3: Cài đặt Google Client Library**

Cài đặt Google Client Library là bước cần thiết để tích hợp các dịch vụ của Google vào ứng dụng .

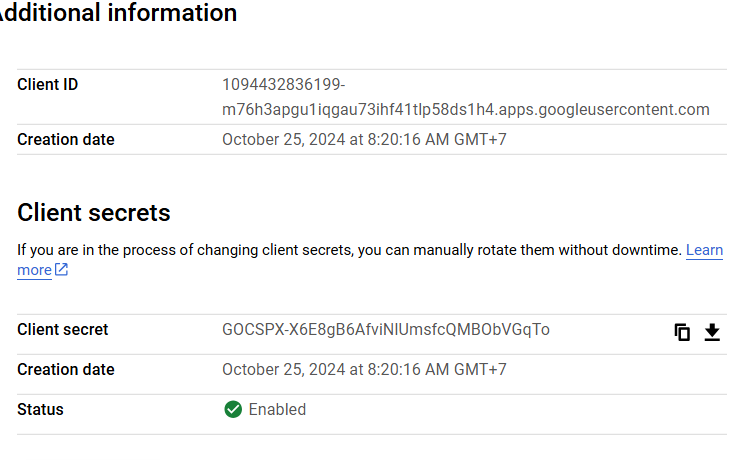
***composer require google/apiclient***

* **Bước 4: Cấu hình Google Client trong PHP**

|  |
| --- |
| <?php  require\_once 'vendor/autoload.php';  $client = new Google\_Client();  $client->setClientId('YOUR\_CLIENT\_ID');  $client->setClientSecret('YOUR\_CLIENT\_SECRET');  $client->setRedirectUri('YOUR\_REDIRECT\_URI');  $client->addScope("email");  $client->addScope("profile");  $googleLoginUrl = $client->createAuthUrl();  ?> |

Tệp ***autoload.php*** trong thư mục vendor của Composer, cho phép sử dụng các thư viện bên ngoài, cụ thể là thư viện Google API Client hỗ trợ cho quá trình đăng nhập.

Các phương thức ***setClientId, setClientSecret, và setRedirectUri*** được sử dụng để cấu hình client với thông tin từ Google Developer Console.



Hình 2. Thông tin Client Id, secret

***$client->addScope('email');***

***$client->addScope('profile');***

Các phương thức này thêm phạm vi quyền mà ứng dụng yêu cầu từ người dùng. Trong trường hợp này, ứng dụng yêu cầu quyền truy cập vào địa chỉ email và thông tin hồ sơ của người dùng.

***$googleLoginUrl = $client->createAuthUrl();***

Tiếp theo sẽ tạo ra một URL mà người dùng sẽ được chuyển hướng đến để đăng nhập vào tài khoản Google của họ và ủy quyền cho ứng dụng.

* **Bước 5: Tạo google\_login xử lý phản hồi từ Google**

Xử lý quá trình xác thực người dùng thông qua Google OAuth 2.0. Sau khi người dùng đăng nhập, mã sẽ lưu trữ thông tin người dùng vào session và cơ sở dữ liệu, đồng thời chuyển hướng người dùng đến trang phù hợp dựa trên vai trò (role) của họ.

|  |
| --- |
| if (isset($\_GET['code'])) {  $token = $client->fetchAccessTokenWithAuthCode($\_GET['code']);  $client->setAccessToken($token['access\_token']);  } |

Kiểm tra xem có mã ủy quyền (code) từ Google không, được gửi qua URL sau khi người dùng đăng nhập thành công. Nếu có mã, nó sẽ được sử dụng để lấy token truy cập thông qua phương thức ***fetchAccessTokenWithAuthCode().*** Token truy cập này được thiết lập cho client, cho phép ứng dụng gửi yêu cầu đến Google API với quyền truy cập của người dùng.

Tiếp theo sẽ lấy thông tin hồ sơ người dùng

|  |
| --- |
| $google\_service = new Google\_Service\_Oauth2($client);  $user\_info = $google\_service->userinfo->get(); |

Tạo một đối tượng dịch vụ Google\_Service\_Oauth2 để tương tác với Google API. Sau đó gọi phương thức ***userinfo->get()*** để lấy thông tin hồ sơ của người dùng, bao gồm các thuộc tính như email, tên. Thông tin này sẽ được sử dụng để xác thực và quản lý người dùng trong ứng dụng.

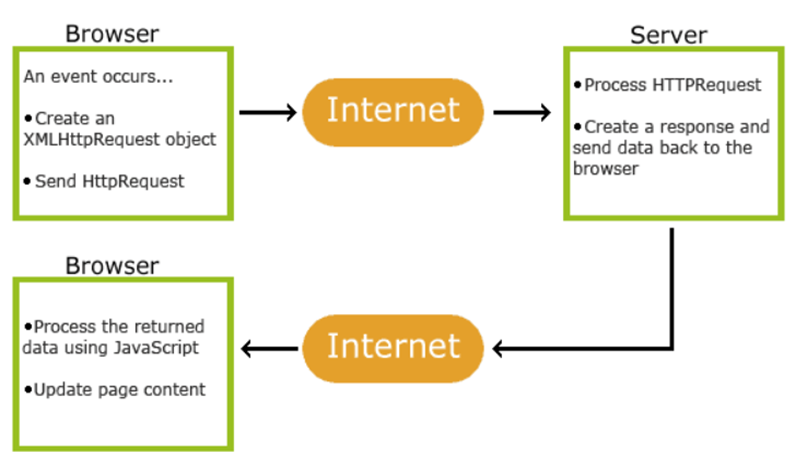
**2.3 Công nghệ AJAX trong PHP**

**2.3.1 Khái niệm AJAX trong PHP**

AJAX viết tắt từ Asynchronous JavaScript and XML, là bộ công nghệ giúp tạo ra các web động hay các ứng dụng giàu tính Internet, cho phép tăng tốc độ ứng dụng web bằng cách cắt nhỏ dữ liệu và chỉ hiển thị những gì cần thiết, thay vì tải đi tải lại toàn bộ trang web, làm như vậy trang web sẽ mượt và đẹp hơn. AJAX không phải một công nghệ đơn lẻ mà là sự kết hợp một nhóm công nghệ với nhau. Trong đó:

* HTML, CSS đóng vai trò hiển thị thông tin, dữ liệu
* Mô hình DOM (Document Object Model) được thực hiện thông qua JavaScript, nhằm hiển thị thông tin động và tương tác với những thông tin được hiển thị
* Đối tượng XMLHttpRequest để trao đổi dữ liệu một cách không đồng bộ với máy chủ web.
* XML thường là định dạng cho dữ liệu truyền.

**2.3.2 Cách hoạt động**



Hình 2. Cách hoạt động của AJAX

1. Một sự kiện xảy ra trong một trang web (trang được tải, một nút được nhấp)

2. Đối tượng XMLHttpRequest được tạo bởi JavaScript

3. Đối tượng XMLHttpRequest gửi yêu cầu đến máy chủ web

4. Máy chủ xử lý yêu cầu

5. Máy chủ gửi phản hồi trở lại trang web

6. Phản hồi được đọc bởi JavaScript

7. Hành động thích hợp (như cập nhật trang) được thực hiện bởi JavaScript.

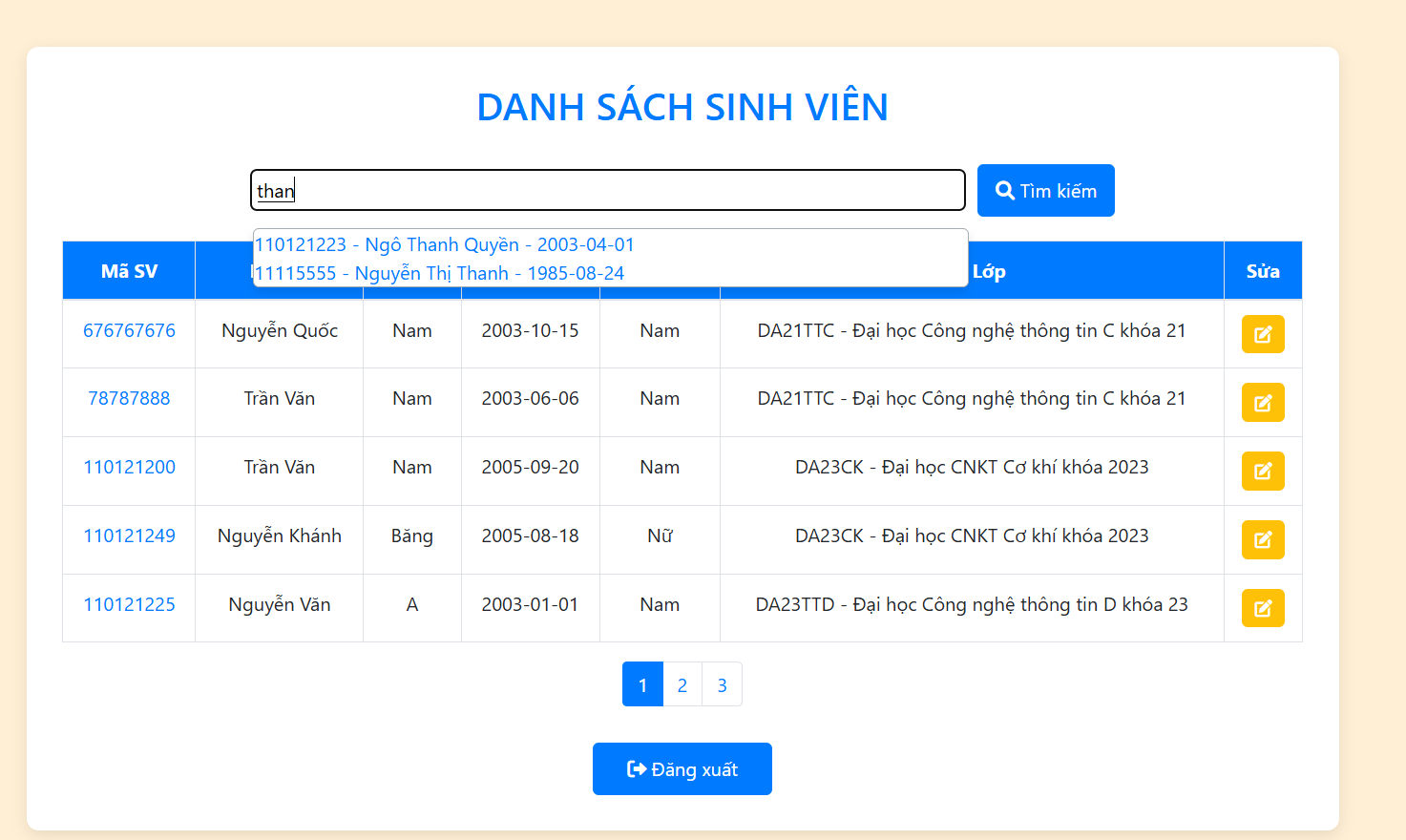
**2.3.3 Ứng dụng của AJAX**

* + - * **Ứng dụng của AJAX trong tìm kiếm trực tiếp (AJAX Livesearch)**

Ví dụ sau đây sẽ minh họa cách tìm kiếm trực tiếp, nơi nhận được kết quả tìm kiếm khi mình nhập.

Tìm kiếm trực tiếp có nhiều lợi ích so với tìm kiếm truyền thống:

* Kết quả được hiển thị khi người dùng nhập
* Kết quả tìm kiếm sẽ thu hẹp khi người dùng tiếp tục nhập
* Nếu kết quả tìm kiếm trở nên quá ít, hãy xóa các ký tự để xem kết quả rộng hơn.

****

Hình 2. Ví dụ ứng dụng AJAX tìm kiếm sinh viên

**Bước 1: Tạo trang HTML với ô tìm kiếm**

Trước tiên, cần một trang HTML với một ô nhập liệu để người dùng nhập từ khóa tìm kiếm:

<input type="text" name="search" class="search-input" placeholder="Tìm kiếm sinh viên..." onkeyup="showResult(this.value)">

<div id="livesearch" class="livesearch"></div>

***onkeyup="showResult(this.value)":*** Đây là sự kiện quan trọng trong tìm kiếm trực tiếp. ***onkeyup*** là một sự kiện JavaScript được kích hoạt mỗi khi người dùng nhấn phím lên sau khi gõ một ký tự trong ô tìm kiếm.

Khi sự kiện này được kích hoạt, hàm JavaScript ***showResult()*** sẽ được gọi và nhận giá trị của ô nhập liệu (giá trị này là từ khóa mà người dùng vừa nhập) thông qua ***this.value.***

Ví dụ: Khi người dùng nhập "A", hàm showResult("A") sẽ được gọi. Khi tiếp tục nhập "Ab", hàm showResult("Ab") sẽ được gọi để cập nhật kết quả tìm kiếm.

**Bước 2: Tạo hàm JavaScript để thực hiện tìm kiếm trực tiếp**

|  |
| --- |
| <script> function showResult(str) {   if (str.length==0){     document.getElementById("livesearch").innerHTML="";    document.getElementById("livesearch").style.border="0px";     return;  }   var xmlhttp=new XMLHttpRequest();  xmlhttp.onreadystatechange=function(){     if (this.readyState==4 && this.status==200){       document.getElementById("livesearch").innerHTML=this.responseText;     document.getElementById("livesearch").style.border="1px solid #A5ACB2";    }  }  xmlhttp.open("GET","livesearch.php?q="+str,true);  xmlhttp.send(); }  </script> |

* Đầu tiên hàm sẽ kiểm tra độ dài của chuỗi nhập liệu (***str.length == 0***): Nếu chuỗi nhập rỗng, sẽ xóa kết quả tìm kiếm và không thực hiện yêu cầu.
* Kế tiếp tạo đối tượng ***XMLHttpRequest***: Đối tượng này cho phép thực hiện yêu cầu HTTP đến máy chủ mà không cần tải lại trang.

***var xmlhttp=new XMLHttpRequest();***

* Tiếp theo sự kiện onreadystatechange:
* Khi trạng thái của ***XMLHttpRequest*** thay đổi, hàm này sẽ kiểm tra xem yêu cầu đã hoàn thành ***(readyState == 4)*** và thành công ***(status == 200).***
* Nếu yêu cầu thành công, nội dung của **<*div id="livesearch">*** trong trang HTML sẽ được cập nhật với phản hồi từ máy chủ ***(this.responseText).***

- Cuối cùng là gửi yêu cầu đến máy chủ:

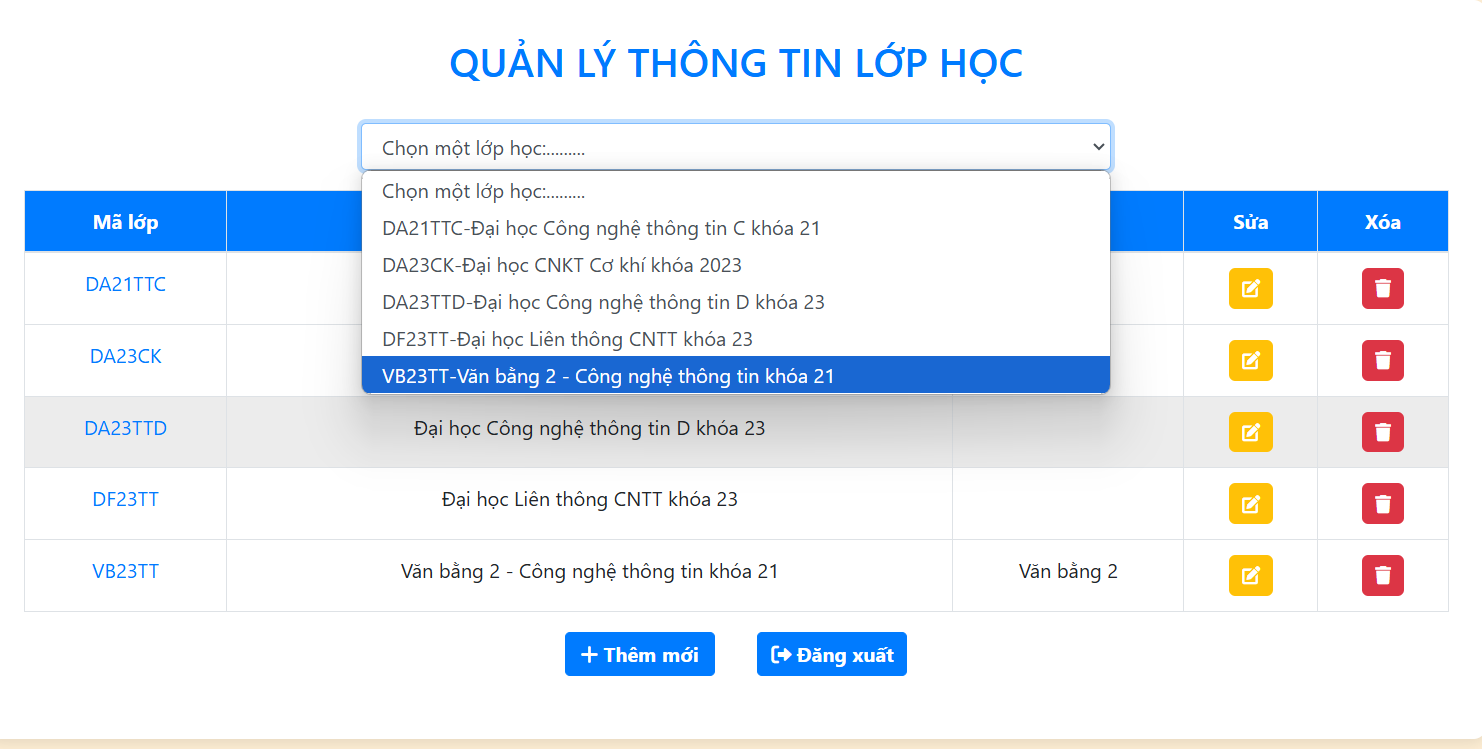
* ***xmlhttp.open("GET","livesearch.php?q="+str,true);*** mở một yêu cầu GET tới tệp ***livesearch.php*** với tham số truy vấn ***q*** là từ khóa tìm kiếm.
* ***xmlhttp.send()*** gửi yêu cầu.

**Bước 3: Tạo tệp PHP (livesearch.php) để xử lý yêu cầu tìm kiếm**

|  |
| --- |
| // Lấy tham số tìm kiếm từ URL  $q isset($\_GET["q"]) ? $\_GET["q"]: '';  if (strlen(string: $q) > 0) {  // Truy văn để tìm kiếm sinh viên theo họ lột, tên sinh viên, hoặc mã sinh viên  $sql = "SELECT FROM sinhvien WHERE holot LIKE '%$q%' OR tenSV LIKE '%$q%' OR maSV LIKE '%$q%' LI  $result = $conn->query(query: $sql);  // Kiểm tra nếu có kết quả  if ($result->num\_rows > 0) {  // Hiển thị kết quả dưới dạng danh sách liên kết  while ($row = $result->fetch\_assoc()) {  echo "<div><a href='thongtinsv.php?maSV=" htmlspecialchars(string: $row['maSV']). "'>  ".htmlspecialchars(string: $row['maSV'])."" htmlspecialchars(string: $row['holot'  -".htmlspecialchars(string: $row['ngaySinh']).  </a></div>";  }  } else {  // Nếu không tìm thấy kết quả nào echo "Không tìm thấy kết quả nào.";  1 |

Ví dụ trên sẽ thực hiện tìm kiếm cơ bản trên cơ sở dữ liệu để trả về danh sách sinh viên dựa trên tham số ***q*** từ URL. Nó tìm kiếm thông tin sinh viên dựa trên các trường: họ lót (hoLot), tên sinh viên (tenSV), mã sinh viên (maSV), giới tính (gioiTinh), và ngày sinh (ngaySinh),.... Kết quả được giới hạn ở 5 sinh viên đầu tiên phù hợp.

* + - * **Ứng dụng của AJAX trong lọc dữ liệu**

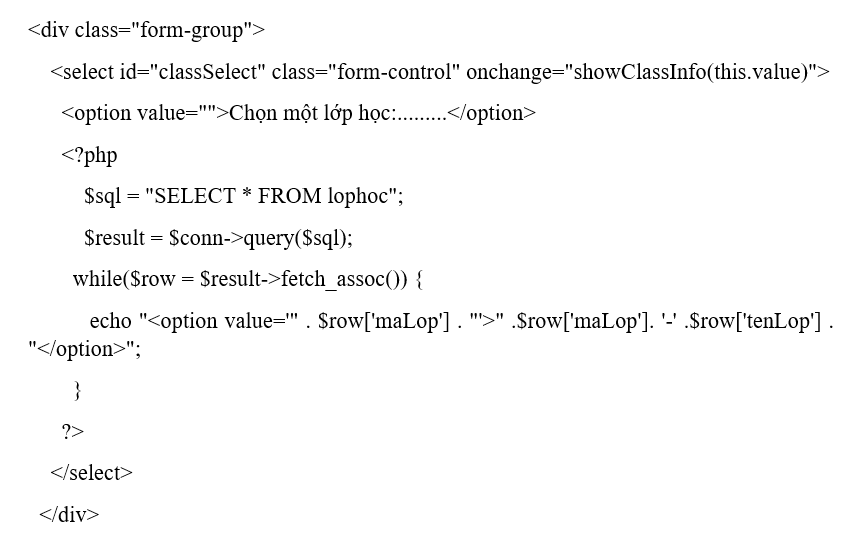
****Ứng dụng của AJAX trong lọc dữ liệu là một cách hiệu quả để cải thiện trải nghiệm người dùng bằng cách tải và hiển thị thông tin mà không cần phải tải lại toàn bộ trang. Dưới đây là ví dụ triển khai AJAX trong một ứng dụng lọc dữ liệu.

Hình 2. Ví dụ ứng dụng AJAX lọc dữ liệu

**Bước 1: Xác định yêu cầu**

Xác định dữ liệu cần lọc như: danh sách sinh viên, lớp học, sản phẩm, v.v.

Tạo giao diện người dùng: Thiết kế giao diện với các thành phần như dropdown, input textbox, hoặc checkbox,... để người dùng có thể chọn hoặc nhập thông tin cần lọc.

**Ví dụ:**

Trong ví dụ trên, khi người dùng chọn một người trong danh sách thả xuống ở trên, một hàm có tên ***showClassInfo()*** là sẽ được thực thi. Chức năng này được kích hoạt bởi sự kiện ***onchange.***

**Bước 2: Viết hàm JavaScript để xử lý sự kiện**

|  |
| --- |
| <script> function showClassInfo(str) {  if (str == "") {  document.getElementById("classInfo").innerHTML = "";  return;  } else {  var xmlhttp = new XMLHttpRequest();  xmlhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  document.getElementById("classInfo").innerHTML = this.responseText;  }  };  xmlhttp.open("GET", "ajax\_get\_class\_info.php?q=" + str, true); // Thay đổi đường dẫn tới file xử lý  xmlhttp.send();  }  }</script> |

* Tương như trong AJAX livesearch, đầu tiên hàm sẽ kiểm tra xem người đó có được chọn không. Nếu không có người nào được chọn ***(str == "")***, hãy xóa nội dung txtHintvà thoát khỏi hàm.
* Kế tiếp tạo đối tượng ***XMLHttpRequest:*** Đối tượng này cho phép thực hiện yêu cầu HTTP đến máy chủ mà không cần tải lại trang.

***var xmlhttp=new XMLHttpRequest();***

* Tiếp theo sự kiện onreadystatechange:
* Khi trạng thái của ***XMLHttpRequest*** thay đổi, hàm này sẽ kiểm tra xem yêu cầu đã hoàn thành ***(readyState == 4)*** và thành công ***(status == 200).***
* Nếu yêu cầu thành công, nội dung của ***<div id="classInfo" class="mt-3"></div>*** trong trang HTML sẽ được cập nhật với phản hồi từ máy chủ ***(this.responseText).***

- Cuối cùng là gửi yêu cầu đến máy chủ:

* ***xmlhttp.open("GET", "ajax\_get\_class\_info.php?q=" + str, true);*** mở một yêu cầu GET tới tệp ***ajax\_get\_class\_info.php*** với tham số truy vấn ***q*** là giá trị đã chọn
* ***xmlhttp.send()*** gửi yêu cầu.

**Bước 3: Tạo tệp PHP *ajax\_get\_class\_info.php*** **để xử lý yêu cầu lọc dữ liệu**

Tương tự như AJAX livesearch, chúng ta sẽ thực hiện những truy vấn cơ bản trên cơ sở dữ liệu để trả về dữ liệu mà ta mong muốn dựa trên tham số ***q*** từ URL.

**2.4 Rewirite**

Việc sử dụng rewrite trong Apache thông qua file ***.htaccess*** có nhiều ý nghĩa quan trọng trong phát triển và quản lý ứng dụng web. Dưới đây là một số lợi ích và ý nghĩa chính của việc sử dụng quy tắc rewrite:

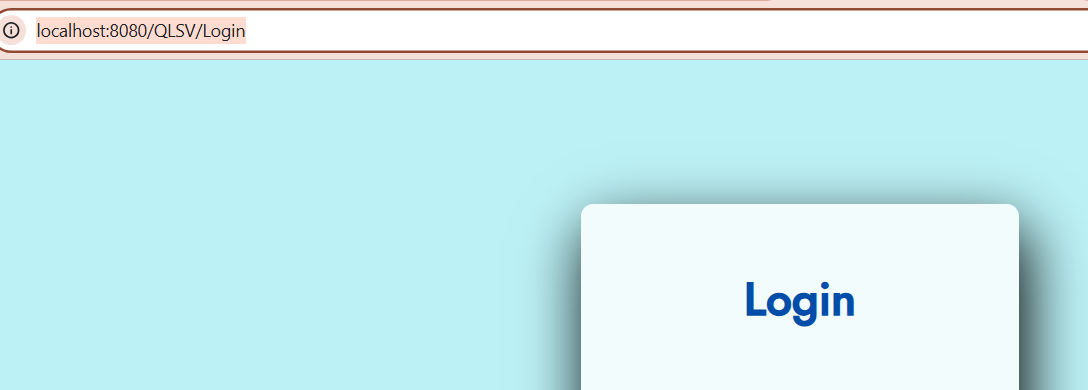
* Các URL được rewrite trở nên dễ đọc và dễ nhớ hơn.
* Bảo mật: Ẩn phần mở rộng file (như .php, .html) giúp giảm thiểu thông tin mà kẻ tấn công có thể thu thập về công nghệ mà bạn đang sử dụng, từ đó bảo vệ ứng dụng tốt hơn.
* Tạo Đường dẫn Động: Với các ứng dụng web động (như PHP), có thể tạo các URL động mà không cần phải chỉ định phần mở rộng file. Điều này giúp cải thiện trải nghiệm người dùng và làm cho ứng dụng trông chuyên nghiệp hơn.

Để sử dụng file ***.htaccess*** với XAMPP nhằm mã hóa URL như trong ví dụ sau, cần thực hiện một số bước thiết lập.

* **Bước 1: Kích hoạt mô-đun mod\_rewrite trong XAMPP**
* Khởi động XAMPP: Nhấp vào nút Config bên cạnh Apache và chọn ***httpd.conf*** để mở file cấu hình.
* Tìm dòng sau: #LoadModule rewrite\_module modules/mod\_rewrite.so. Sau đó bỏ dấu # ở đầu dòng để kích hoạt mô-đun mod\_rewrite
* Thêm hoặc sửa dòng sau: AllowOverride All
* **Bước 2: Tạo file .htaccess**

|  |
| --- |
| # Bật chế độ viết lại  RewriteEngine On  # Đặt đường dẫn cơ sở  RewriteBase /QLSV/  # Ẩn phần mở rộng .php cho trang Login  RewriteRule ^Login$ Login.php [L]  # Ẩn phần mở rộng .php cho trang lop  RewriteRule ^lophoc$ lophoc.php [L] |

Sau khi cấu hình xong, ta có thể chạy mà không cần phần đuôi mở rộng (.php)

****

Hình 2. Ví dụ Rewirite trang login.php

**2.5 Ứng dụng Javascript gọi API hiển thị địa lý**

* **Ưu điểm**
* Trải nghiệm người dùng tốt: Việc tải dữ liệu động giúp người dùng có thể chọn địa chỉ một cách dễ dàng và tiện lợi, mà không cần phải tải lại toàn bộ trang.
* Hiệu suất cao: Ứng dụng chỉ tải dữ liệu cần thiết theo yêu cầu, giúp giảm bớt tải cho server và cải thiện tốc độ phản hồi.
* **Hạn chế**
* Phụ thuộc vào API bên ngoài: Nếu API ngừng hoạt động hoặc có thay đổi về cấu trúc dữ liệu, ứng dụng có thể gặp lỗi.
* Xử lý lỗi đơn giản: Hiện tại, ứng dụng chỉ hiển thị thông báo lỗi cơ bản khi không tải được dữ liệu. Nên thêm tính năng thông báo chi tiết hơn cho người dùng để cải thiện trải nghiệm khi gặp lỗi.

Sau đây là ví dụ việc ứng dụng Javascript gọi API hiển thị địa lý **:**

* **Gọi API tải danh sách địa lý Việt Nam**

Ứng dụng này sử dụng API từ địa chỉ ***"https://provinces.open-api.vn/api/"*** để lấy thông tin về các địa danh. Mỗi loại địa danh được tải bằng các hàm khác nhau:

|  |
| --- |
| const host = "https://provinces.open-api.vn/api/"; |

**callAPI():** Gọi API để tải danh sách các tỉnh/thành phố.

**callApiDistrict():** Gọi API để tải danh sách các quận/huyện dựa trên mã tỉnh/thành phố được chọn.

**callApiWard():** Gọi API để tải danh sách các phường/xã dựa trên mã quận/huyện được chọn.

|  |
| --- |
| Ví dụ:  // Gọi API để tải danh sách tỉnh/thành phố  const callAPI = (api) => {  return axios.get(api)  .then((response) => {  renderData(response.data, "province");  })  .catch((error) => {  console.error("Error loading provinces:", error);  alert("Có lỗi khi tải thông tin tỉnh/thành phố.");  });  }; |

Mỗi hàm này đều sử dụng thư viện Axios để gửi yêu cầu GET tới API và xử lý kết quả trả về. Khi dữ liệu tải thành công, chúng sẽ được chuyển đến hàm ***renderData()*** để hiển thị vào các dropdown (ô chọn).

|  |
| --- |
| // Hàm để hiển thị dữ liệu vào dropdown  const renderData = (array, select) => {  let row = '<option disabled selected value="">Chọn</option>';  array.forEach(element => {  row += `<option value="${element.code}">${element.name}</option>`;  });  const selectElement = document.getElementById(select);  selectElement.innerHTML = row;  selectElement.disabled = false; // Kích hoạt dropdown khi có dữ liệu  };  // Sự kiện khi chọn tỉnh/thành phố  document.getElementById("province").addEventListener("change", function() {  callApiDistrict(host + "p/" + this.value + "?depth=2");  document.getElementById("district").disabled = true; // Vô hiệu hóa dropdown quận/huyện  document.getElementById("ward").disabled = true; // Vô hiệu hóa dropdown phường/xã }); |

Hàm ***renderData()*** có nhiệm vụ hiển thị dữ liệu vào các dropdown tương ứng (tỉnh/thành phố, quận/huyện, hoặc phường/xã). Nó nhận vào hai tham số:

* array: Mảng dữ liệu từ API trả về.
* select: Tên của dropdown cần hiển thị dữ liệu.
* **Tạo nội dung cho dropdown:**

Biến ***row*** chứa một ***<option>*** mặc định với giá trị rỗng, để người dùng có thể chọn. Sử dụng ***forEach*** để duyệt qua từng phần tử trong mảng array, tạo một ***<option>*** cho mỗi phần tử với giá trị code và nội dung hiển thị là name.

* **Cập nhật nội dung của dropdown**

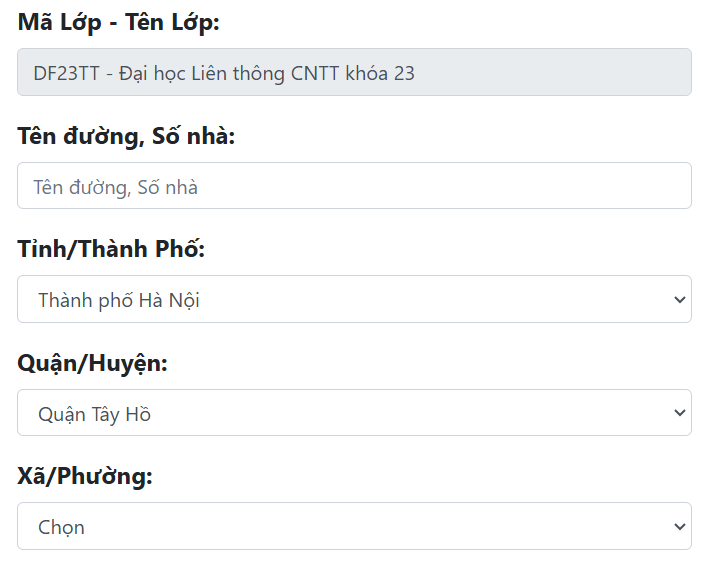
Tìm phần tử HTML bằng ***document.getElementById(select)*** và cập nhật nội dung của nó bằng ***row***. Kích hoạt ***dropdown (disabled = false)*** để có thể chọn được.

* **Sự kiện khi chọn tỉnh/thành phố**

Phần này lắng nghe sự kiện "change" khi người dùng chọn một tỉnh/thành phố:

* Gọi API để lấy dữ liệu quận/huyện: Sử dụng ***callApiDistrict*** để gọi API với URL chứa mã của tỉnh/thành phố đã chọn và độ sâu dữ liệu là 2 (bao gồm cả quận/huyện).
* Vô hiệu hóa dropdown quận/huyện và phường/xã:

Trước khi có dữ liệu mới, hai dropdown này sẽ bị vô hiệu hóa (disabled = true) để tránh việc chọn sai dữ liệu.



Hình 2. Ví dụ lấy địa chỉ từ API

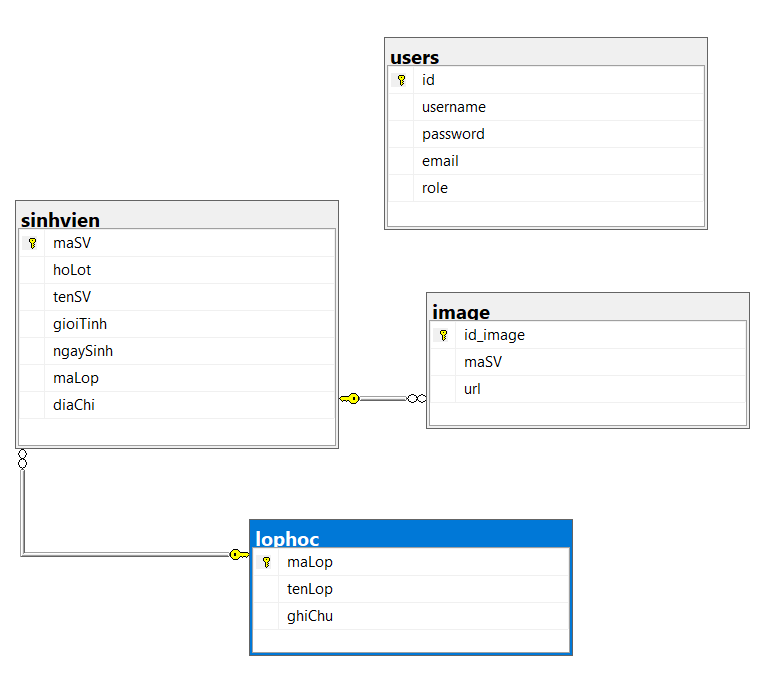
**CHƯƠNG 3: THỰC HIỆN HÓA NGHIÊN CỨU**

**3.1 Thiết kế cơ sở dữ liệu**

**3.1.1 Cấu trúc cơ sở dữ liệu**

* **Bảng lophoc: Lưu thông tin về các lớp học.**
* maLop: Mã lớp (khóa chính).
* tenLop: Tên lớp.
* ghiChu: Ghi chú về lớp học.
* **Bảng sinhvien: Lưu thông tin sinh viên.**
* maSV: Mã sinh viên (khóa chính).
* hoLot: Họ lót của sinh viên.
* tenSV: Tên sinh viên.
* gioiTinh: Giới tính.
* ngaySinh: Ngày sinh của sinh viên.
* maLop: Mã lớp (khóa ngoại, liên kết với bảng lophoc).
* diaChi: Địa chỉ của sinh viên.
* **Bảng users: Quản lý thông tin người dùng.**
* id: ID người dùng (khóa chính, tự động tăng).
* username: Tên đăng nhập.
* password: Mật khẩu.
* role: Vai trò của người dùng (admin hoặc user).
* email: email người dùng
* **Bảng image: Lưu thông tin ảnh của sinh viên.**
* id\_image: ID ảnh (khóa chính, tự động tăng).
* maSV: Mã sinh viên (khóa ngoại, liên kết với bảng sinhvien).
* url: Đường dẫn ảnh.

**3.1.2 Mô hình ràng buộc cơ sở dữ liệu**

****

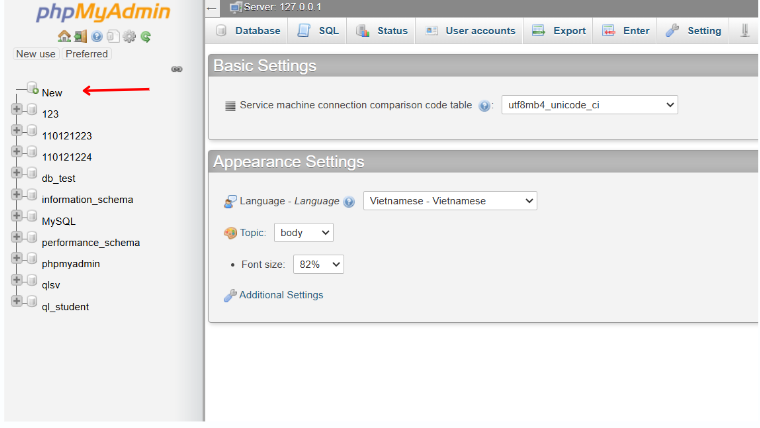
Hình 3. Mô hình ràng buộc cơ sở dữ liệu

**3.2 Kết nối cơ sở dữ liệu**

**3.2.1 Tạo CSDL**

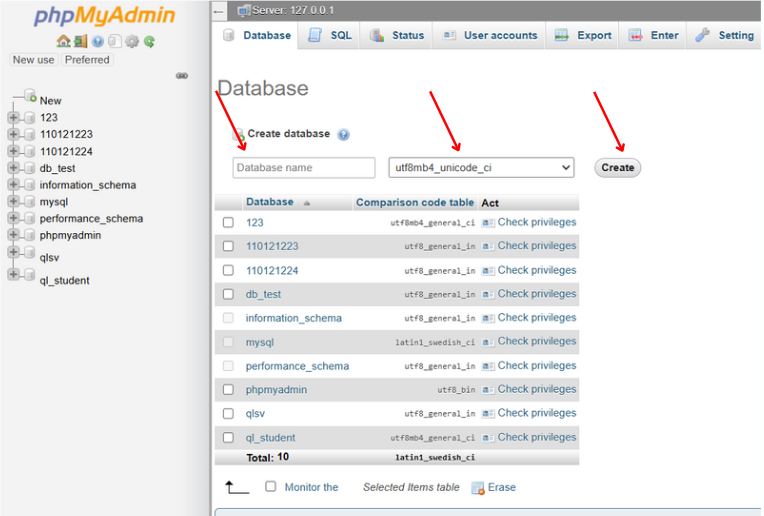
1. **Tạo bằng thủ công**

* Chạy xampp vào trang phpMyAdmin, sau đó chọn new để tạo một csdl mới



Hình 2. Tạo thủ công cơ sở dữ liệu mới

* Nhập tên csdl muốn tạo vào Database name và chọn ngôn ngữ sau đó bấm Create



Hình 2. Nhập tên csdl mới

1. **Tạo bằng lệnh**



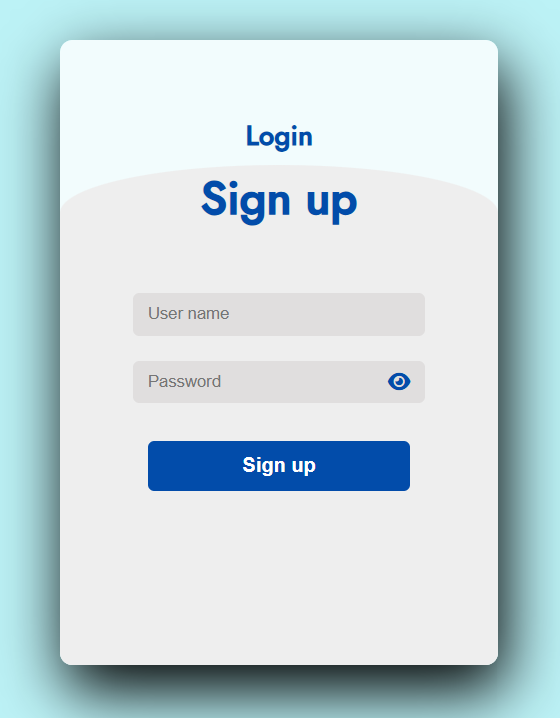
Hình 2. Tạo csdl bằng lệnh

**3.2.2 Kết nối cơ sở dữ liệu**

Kết nối csdl bằng các lệnh sau:

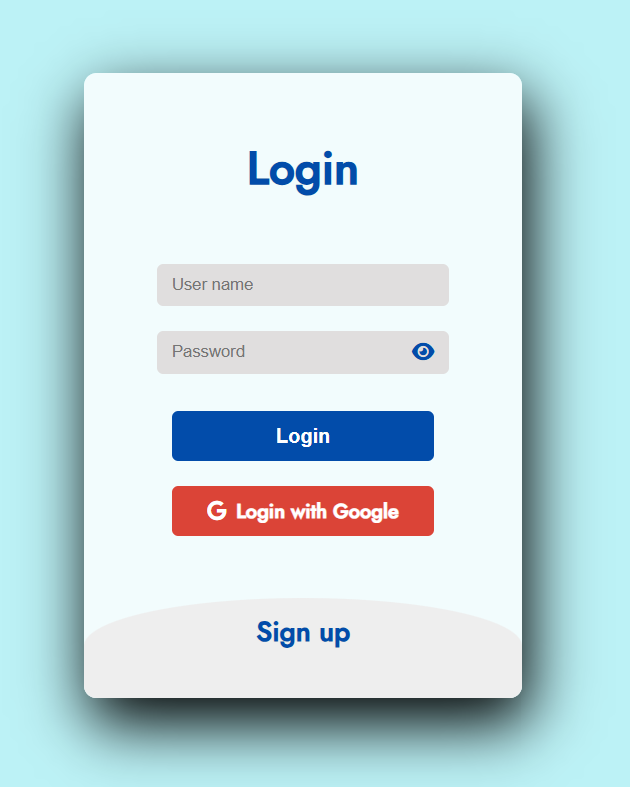
|  |
| --- |
| <?php  //connect.php  $servername = "localhost"; // Địa chỉ máy chủ cơ sở dữ liệu (hoặc máy chủ cục bộ)  $username = "root"; // Tên đăng nhập cơ sở dữ liệu  $password = ""; // Mật khẩu cơ sở dữ liệu  $dbname = "db\_test"; // Tên cơ sở dữ liệu  // Tạo kết nối  $conn = new mysqli($servername, $username, $password, $dbname);  // Kiểm tra kết nối  if ($conn->connect\_error) { die("Connection failed: ".$conn->connect\_error);  }?> |

**CHƯƠNG 4: XÂY DỰNG HỆ THỐNG**

**4.1 Đăng ký**

Hình 4. Đăng ký

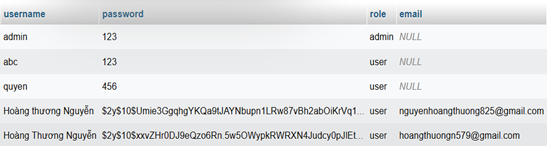
Người dùng sẽ nhập tên tài khoản vào trường ***username*** và mật khẩu vào trường ***password*** trong form đăng ký. Sau khi nhập xong, hệ thống sẽ lưu trữ thông tin tên tài khoản và mật khẩu vào cơ sở dữ liệu để sử dụng cho quá trình đăng nhập sau này.

**4.2 Đăng nhập**

Hình 4. Đăng nhập

Khi đến với form đăng nhập, người dùng sẽ nhập tên tài khoản và mật khẩu đã sử dụng trong quá trình đăng ký. Hệ thống sẽ kiểm tra thông tin này với dữ liệu đã lưu trong cơ sở dữ liệu. Nếu tên tài khoản và mật khẩu khớp với thông tin đã đăng ký, người dùng sẽ đăng nhập thành công. Ngược lại, nếu thông tin không khớp, quá trình đăng nhập sẽ thất bại.

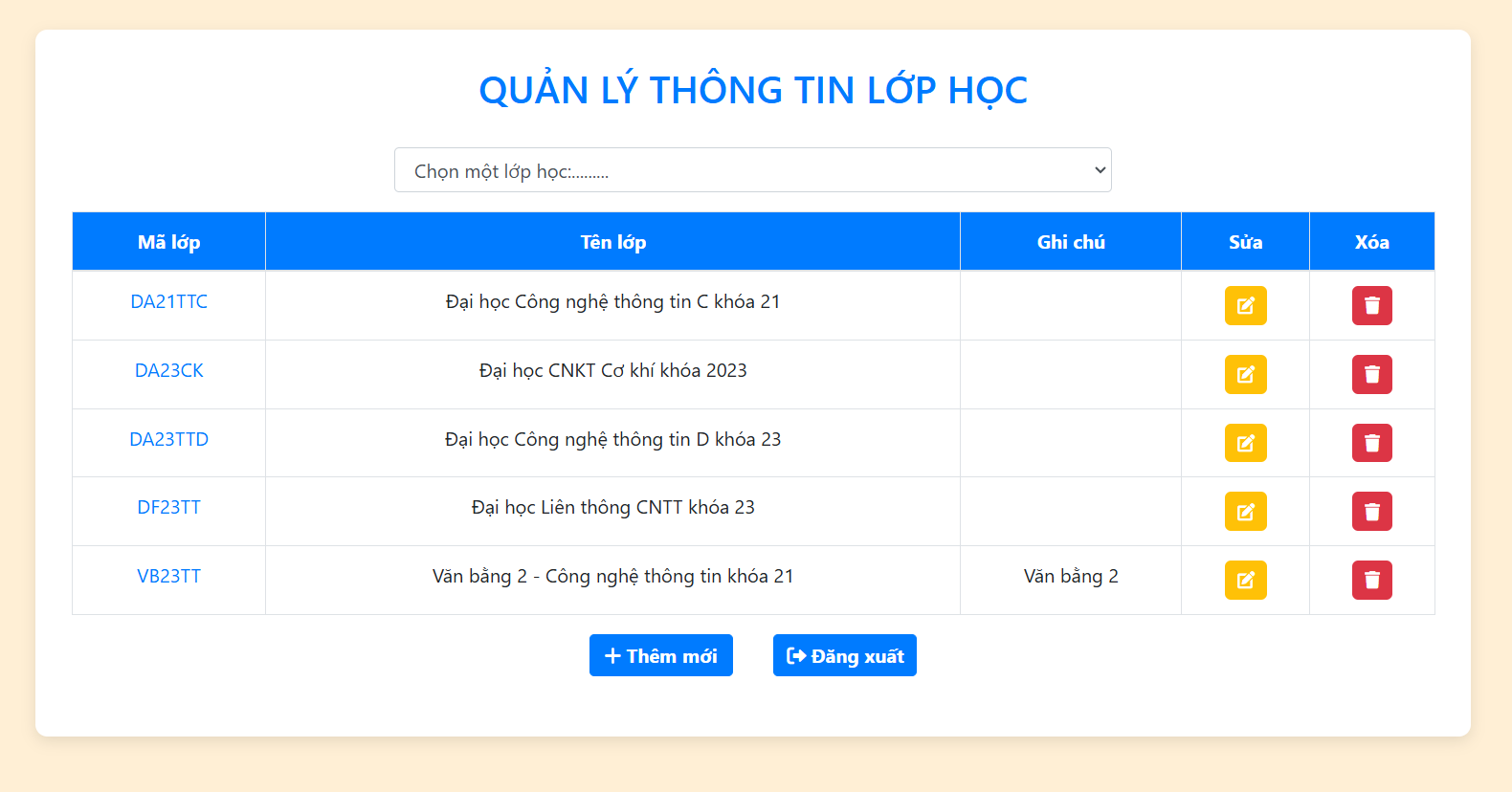
Khi người dùng đăng nhập bằng tài khoản Google, hệ thống sẽ tự động trích xuất thông tin như username và email từ tài khoản của họ và lưu trữ vào cơ sở dữ liệu.



Hình 4. Thông tin người dùng Google đã lưu thành công

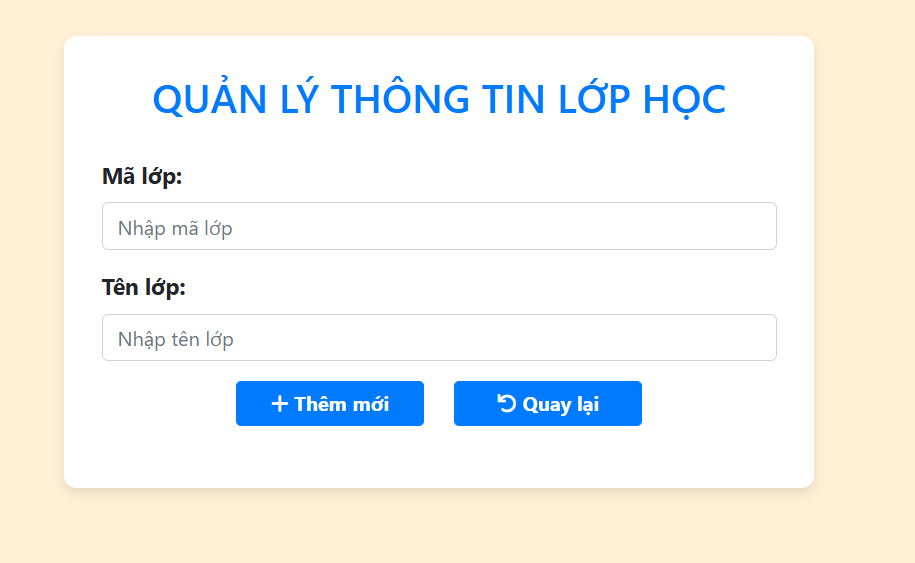
**4.3 Phân quyền**

Sau khi đăng nhập thành công, hệ thống sẽ kiểm tra vai trò (role) của người dùng. Nếu người dùng có vai trò là admin, họ sẽ được chuyển đến giao diện quản trị với các chức năng đầy đủ như sau:



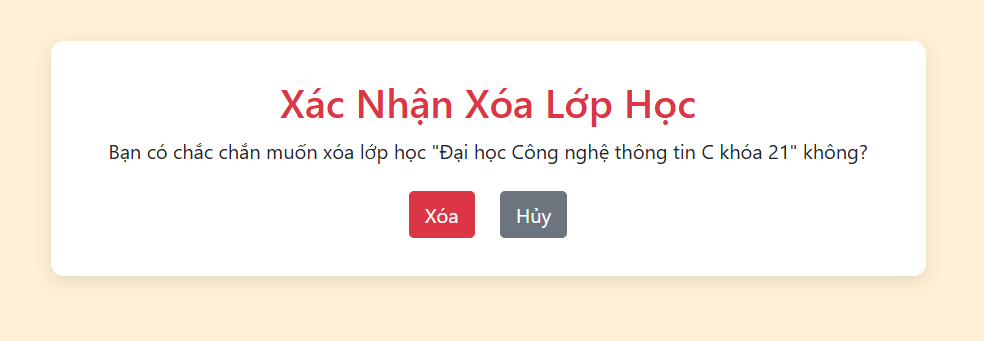
Hình 4. Quyền admin

* Đối với quyền admin, họ sẽ có quyền truy cập vào các chức năng quản lý sau:
* Quản lý thông tin lớp học:
* Thêm lớp học: Cho phép admin tạo mới các lớp học với thông tin chi tiết như tên lớp, mã lớp sao đó nhấn nút thêm mới dữ liệu vừa thêm sẽ được insert vào csdl.

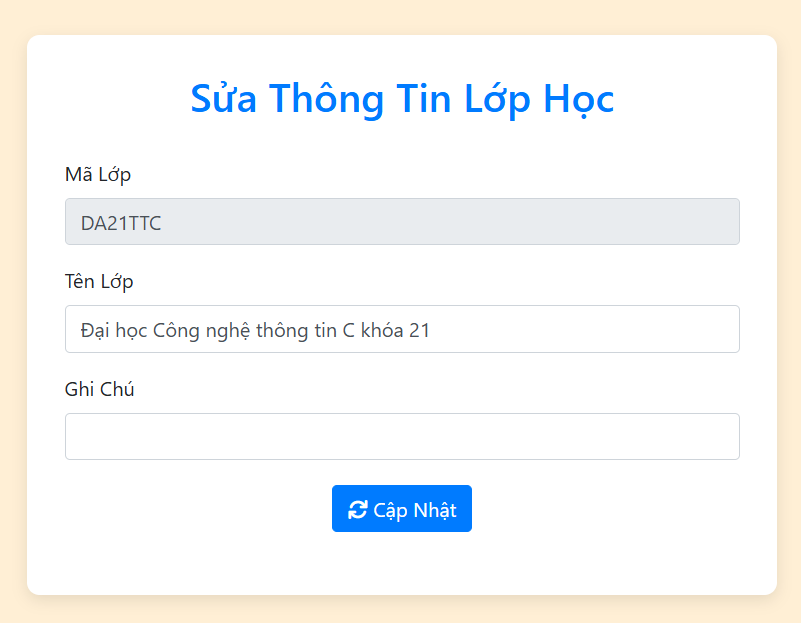


Hình 4. Thêm lớp học mới

* Xóa lớp học: Cho phép admin xóa các lớp học không còn hoạt động hoặc không cần thiết. Khi xóa lớp sẽ xóa hết tất cả sinh viên thuộc lớp đó

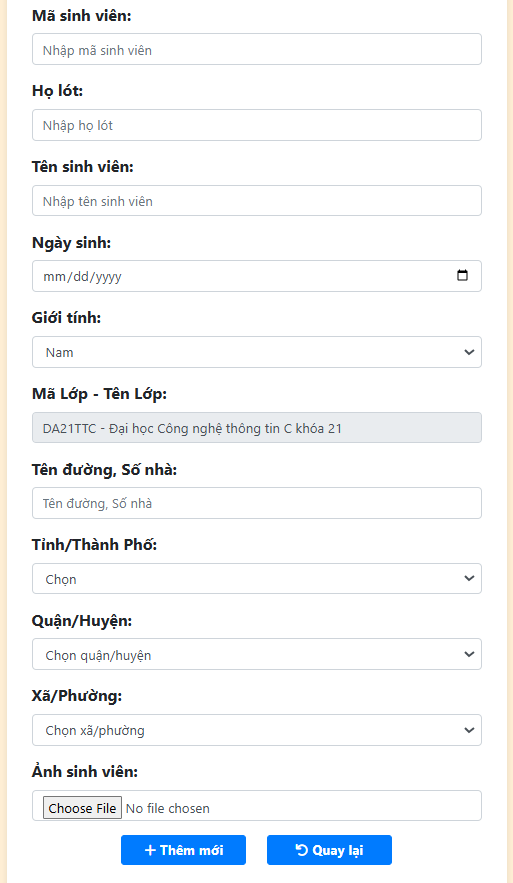


Hình 4. Xóa lớp học

* Sửa lớp học: Cho phép admin chỉnh sửa thông tin của các lớp học hiện có.

Hình 4. Sửa thông tin lớp học

* Quản lý thông tin sinh viên:
* Thêm sinh viên: Cho phép admin thêm mới thông tin sinh viên vào hệ thống, bao gồm tên, mã sinh viên, lớp học, và các thông tin khác. Khi thêm phải đảm bảo các điều kiện sau: mã sinh viên phải là số, ngày sinh phải đảm bảo sinh viên đủ 18 tuổi trở lên, ảnh sinh viên phải là file ảnh với định dạng JPG, JPEG, PNG & GIF, và không được vượt quá 500kb



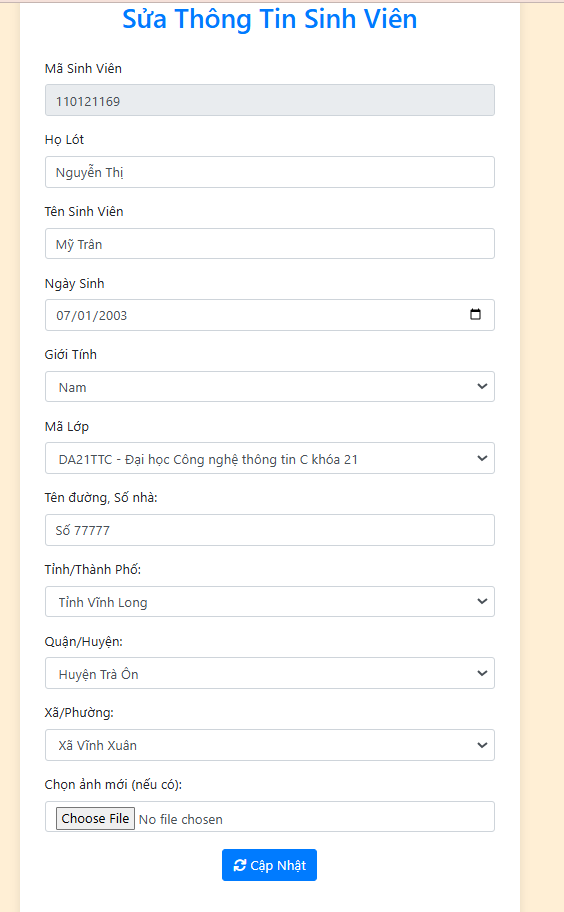
Hình 4. Thêm sinh viên

* Xóa sinh viên: Cho phép admin xóa thông tin sinh viên khi cần thiết, chẳng hạn như khi sinh viên tốt nghiệp hoặc rời khỏi trường.



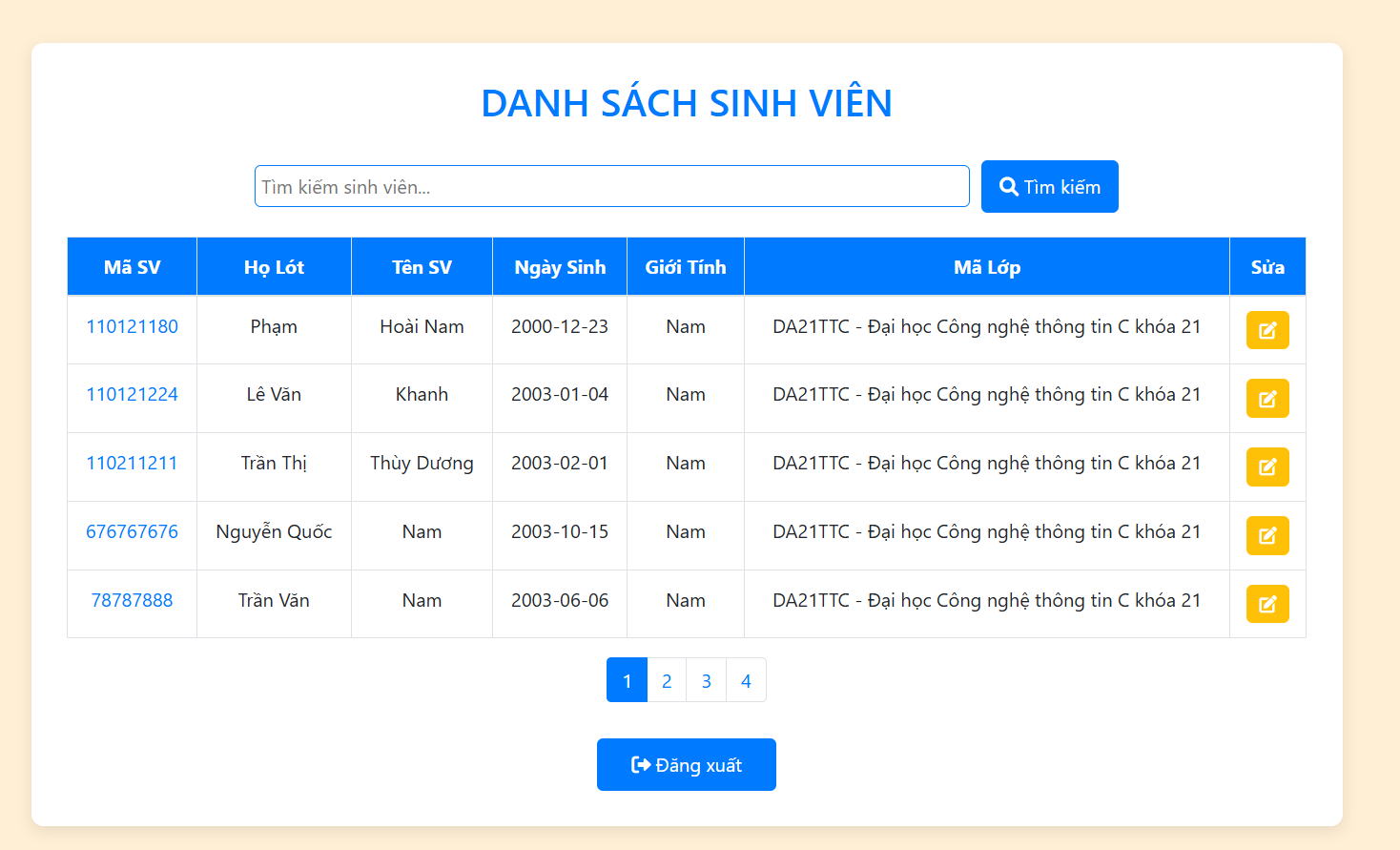
Hình 4. Xóa sinh viên

* Sửa sinh viên: Cho phép admin chỉnh sửa thông tin sinh viên hiện có.

****

Hình 4. Sửa thông tin sinh viên

* Đối với vai trò là user, giao diện sẽ được thiết kế đơn giản và tập trung vào các chức năng cơ bản là xem và sửa thông tin sinh viên.



Hình 4. Quyền user

****

Hình 4. Xem thông tin sinh viên

**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1 Kết luận**

**5.1.1 Kết quả đạt được**

Hệ thống quản lý sinh viên được xây dựng dựa trên nền tảng PHP và MySQL đã hoàn thành đầy đủ các chức năng cơ bản như thêm, sửa, xóa cho việc quản lý thông tin sinh viên, lớp học. Giao diện người dùng được thiết kế thân thiện, dễ sử dụng.

Trong quá trình phát triển, nhóm tìm hiểu và áp dụng nhiều công nghệ và kỹ thuật như AJAX để nâng cao trải nghiệm người dùng, cho phép tải dữ liệu mà không cần làm mới toàn bộ trang. Hệ thống còn tích hợp việc lấy địa chỉ từ API để cung cấp thông tin chính xác hơn cho người dùng. Bên cạnh đó, tính năng phân trang giúp quản lý thông tin một cách hiệu quả và dễ dàng, đồng thời việc hỗ trợ đăng nhập qua Google mang lại sự thuận tiện cho người dùng trong việc truy cập hệ thống.

**5.1.2 Hạn chế**

* Giao diện chưa thật sự hoàn hảo, các chức năng còn hạn chế chưa thể đáp ứng đầy đủ nhu cầu của người dùng
* Cơ sở dữ liệu ít chưa đáp ứng được nhu cầu thực tế

**5.2 Hướng phát triển**

* Thiết kế lại giao diện để trở hoàn hảo hơn đáp ứng mọi nhu cầu của người dùng bằng cách sử dụng các công nghệ, kỹ thuật hiện đại hơn.
* Thêm các chức năng mới để đáp ứng đầy đủ nhu cầu của người dùng
* Nghiên cứu và áp dụng Framework cho hệ thống để tối ưu hóa quy trình phát triển, giảm thiểu lỗi và tăng cường khả năng bảo trì mã nguồn.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Luke Welling and Laura Thomson. PHP and MySQL Web Development Fifth Edition.
2. W3schools.com. PHP Tutorial
3. Id.VinaStar.net. Viết lại đường dẫn URL mod\_rewrite .htaccess Apache.