**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

###### KHOA ĐIỆN TỬ - VIỄN THÔNG

##### TRẦN QUÂN

### 

**XÂY DỰNG TRANG WEB QUẢN LÝ THÔNG TIN CỦA GROUP CÁC CÂU LẠC BỘ TRÊN FACEBOOK.**

**SEMINAR TỐT NGHIỆP CỬ NHÂN**

**NGÀNH KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG**

**CHUYÊN NGÀNH: VIỄN THÔNG – MẠNG**

##### TP. Hồ Chí Minh – Năm 2023

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA ĐIỆN TỬ - VIỄN THÔNG

**SEMINAR TỐT NGHIỆP CỬ NHÂN**

**NGÀNH KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG**

**CHUYÊN NGÀNH: VIỄN THÔNG – MẠNG**

**XÂY DỰNG TRANG WEB QUẢN LÝ THÔNG TIN CỦA GROUP CÁC CÂU LẠC BỘ TRÊN FACEBOOK.**

**Họ và tên sinh viên: Trần Quân**

**Mã số SV: 19200443**

NGƯỜI HƯỚNG DẪN KHOA HỌC:

Ths. Trương Tấn Quang

TP. Hồ Chí Minh – Năm 2023

**XÂY DỰNG TRANG WEB QUẢN LÝ THÔNG TIN CỦA GROUP CÁC CÂU LẠC BỘ TRÊN FACEBOOK.**

**BUILDING THE INFORMATION MANAGEMENT SITE OF THE GROUP OF CLUBS ON FACEBOOK**

**Sinh viên: Trần Quân**

**Người hướng dẫn: Ths. Trương Tấn Quang**

**Hội đồng đánh giá:**

1. Chủ tịch: xác nhận:
2. Chủ tịch: xác nhận:
3. Adc
4. Abc
5. Abc

# LỜI CẢM ƠN

Em muốn bày tỏ lòng biết ơn chân thành đến tất cả các thầy cô trong khoa Điện tử - Viễn thông đã đồng hành và dạy dỗ em trong suốt quãng đường đại học. Em muốn gửi lời cảm ơn đặc biệt tới Ban chủ nhiệm khoa vì đã tạo điều kiện học tập tốt nhất cho chúng em, cả về cơ sở vật chất và chất lượng giảng dạy. Em cũng muốn bày tỏ lòng biết ơn đến tất cả các thầy cô chuyên ngành Viễn thông - Mạng đã hỗ trợ và chỉ dẫn chúng em khi chúng em bước vào chuyên ngành này.

Em muốn gửi lời cảm ơn đặc biệt tới thầy Đặng Lê Khoa, trưởng Bộ môn Viễn thông Mạng, vì đã tạo điều kiện và cung cấp cho chúng em nhiều cơ hội học tập, giảng dạy và chia sẻ những kinh nghiệm quý báu. Thầy đã luôn ủng hộ, giúp đỡ và động viên chúng em để chúng em có thể có nhận thức đúng đắn và xác định hướng đi cho tương lai của mình.

Đặc biệt, em muốn bày tỏ lòng biết ơn đến thầy Trương Tấn Quang và anh Nguyễn Mạnh Tiến vì đã dành thời gian và tâm huyết để hướng dẫn, hỗ trợ và chia sẻ những kiến thức quý báu với chúng em. Sự phản hồi tỉ mỉ và chỉ bảo của hai người đã giúp em xây dựng tiểu luận trong thời gian qua. Một lần nữa, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Trương Tấn Quang và anh Nguyễn Mạnh Tiến.

Cuối cùng, em xin kính chúc khoa Điện tử - Viễn thông ngày càng phát triển vững mạnh. Tiểu luận của em còn những hạn chế về năng lực và những thiếu sót trong quá trình nghiên cứu. Em xin lắng nghe và tiếp thu những ý kiến của giáo viên phản biện để hoàn thiện, bổ sung kiến thức. Vì vậy những đóng góp quý báu từ quý thầy cô sẽ giúp em khắc phục những sai sót ấy và có thể hoàn thiện đề tài nghiên cứu tốt hơn. Em xin cảm ơn quý thầy cô và kính chúc các thầy cô thật nhiều sức khỏe!

A picture containing darkness, black, night

Description automatically generated

*Tp.Hồ Chi Minh, ngày 1 tháng 7 năm 2023*

Người viết

**Trần Quân**

# TÓM TẮT

Trang web này được thiết kế để hiển thị thông tin về bài đăng và thành viên của một CLB. Sử dụng Python và cơ sở dữ liệu MySQL, dữ liệu bài đăng được đồng bộ và lưu trữ. Giao diện trang web được thiết kế bằng HTML và CSS để tạo ra một khung chức năng hấp dẫn.

Sử dụng framework Laravel, trang web có khả năng lấy thông tin từ cơ sở dữ liệu và hiển thị chúng lên trang web. Người dùng có thể truy cập vào danh sách bài đăng và thông tin thành viên của CLB.

Trang web cũng có các chức năng bổ sung như tìm kiếm và lọc để người dùng dễ dàng tìm kiếm và lọc bài đăng theo tiêu đề hoặc các tiêu chí khác.

Trang web đã được tối ưu hóa và kiểm tra để đảm bảo tính năng hoạt động ổn định và giao diện phản hồi tốt trên các thiết bị và kích thước màn hình khác nhau.

Tóm lại, trang web này cung cấp một giao diện hấp dẫn cho người dùng hiển thị thông tin về bài đăng và thành viên CLB. Có tính năng tìm kiếm và lọc để giúp người dùng dễ dàng tìm kiếm nội dung mong muốn trên trang web này.

# ABSTRACT

This website is designed to display information about posts and members of a club. Using Python and MySQL database, post data is synchronized and stored. The website interface is designed using HTML and CSS to create an appealing functional framework.

Using the Laravel framework, the website is capable of retrieving information from the database and displaying it on the website. Users can access the list of posts and information about club members.

The website also includes additional features such as search and filtering to allow users to easily find and filter posts based on titles or other criteria.

The website has been optimized and tested to ensure stable functionality and a responsive interface across different devices and screen sizes.

In summary, this website provides an attractive interface for users to display information about club posts and members. It includes search and filtering features to facilitate easy search for desired content on the website.

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc139369281)

[TÓM TẮT ii](#_Toc139369282)

[ABSTRACT ii](#_Toc139369283)

[DANH SÁCH CHỮ VIẾT TẮT 4](#_Toc139369284)

[DANH MỤC CÁC HÌNH 5](#_Toc139369285)

[DANH MỤC CÁC BẢNG 8](#_Toc139369286)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 9](#_Toc139369287)

[1.1 Giới thiệu 9](#_Toc139369288)

[1.2 Lý do chọn đề tài 9](#_Toc139369289)

[1.3 Mục tiêu đề tài 10](#_Toc139369290)

[1.4 Phương pháp nghiên cứu 10](#_Toc139369291)

[1.5 Kết luận: 11](#_Toc139369292)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 12](#_Toc139369293)

[2.1 Giới thiệu 12](#_Toc139369294)

[2.1.1 . HTML 12](#_Toc139369295)

[2.1.2 . CSS 12](#_Toc139369296)

[2.1.3 Python 12](#_Toc139369297)

[2.1.4 Ngôn ngữ PHP 12](#_Toc139369298)

[2.1.5 Framework Laravel 12](#_Toc139369299)

[2.2 Phân tích kỹ thuật 13](#_Toc139369300)

[2.2.1 Cài đặt Xampp 13](#_Toc139369301)

[2.2.2 Cài đặt Composer 16](#_Toc139369302)

[2.2.3 Lập trình cơ sở dữ liệu MySQL 20](#_Toc139369303)

[2.2.4 Cài đặt PHP 21](#_Toc139369304)

[2.2.5 Cài đặt Framework Laravel và mô hình MVC 26](#_Toc139369305)

[2.2.6 Cài đặt Python và thư viện facebook\_scraper. 29](#_Toc139369306)

[2.3 Kết luận 33](#_Toc139369307)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 34](#_Toc139369308)

[3.1 Phân tích mối quan niệm dữ liệu: 34](#_Toc139369309)

[3.1.1 Mô tả thực thể kết hợp diễn đạt các thành phần dữ liệu ở mức quan niệm dựa trên bảng danh mục các dữ liệu cơ bản. 34](#_Toc139369310)

[3.1.2 Mô tả các mối kết hợp 35](#_Toc139369311)

[3.1.3 Từ điển cơ sở dữ liệu 36](#_Toc139369312)

[3.2 Mô hình quan hệ (Relational Data Diagram ) 39](#_Toc139369313)

[3.3 Dùng thư viện facebook\_scraper để đồng bộ dữ liệu từ group về MySQL 40](#_Toc139369314)

[3.4 Dùng mô hình MVC trong Laravel để lấy dữ liệu từ MySQL về giao diện. 54](#_Toc139369315)

[3.4.1 Taọ Model 54](#_Toc139369316)

[3.4.2 Tạo Controller 56](#_Toc139369317)

[3.4.3 Tạo Views 58](#_Toc139369318)

[3.5 Kết luận 58](#_Toc139369319)

[CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 59](#_Toc139369320)

[4.1 Tổ chức 59](#_Toc139369321)

[4.1.1 Database 59](#_Toc139369322)

[4.1.2 Website 59](#_Toc139369323)

[4.2 Kết quả 60](#_Toc139369324)

[4.2.1 Trang chủ: 60](#_Toc139369325)

[4.2.2 Trang chi tiết bài viết: 61](#_Toc139369326)

[4.2.3 Trang giới thiệu câu lạc bộ 62](#_Toc139369327)

[4.3 Kết luận 62](#_Toc139369328)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 63](#_Toc139369329)

[5.1 Kết luận 63](#_Toc139369330)

[5.2 Hướng phát triển 63](#_Toc139369331)

[TÀI LIỆU THAM KHẢO 65](#_Toc139369332)

# DANH SÁCH CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| PHP | Personal Home Page |
| WEB | Website |
| CLB | Câu lạc bộ |
| VSC | Visual Studio Code |
| ORM | Object Relational Mapping |
| W3C | World Wide Web Consortium |

# DANH MỤC CÁC HÌNH

[Hình 2.2.1‑1: Bước 1 cài đặt Xampp 14](#_Toc139473880)

[Hình 2.2.1‑2: Bước 2 cài đặt Xampp 14](#_Toc139473881)

[Hình 2.2.1‑3: Bước 3 Cài đặt Xampp 15](#_Toc139473882)

[Hình 2.2.1‑4: Bước 4 cài đặt Xampp 15](#_Toc139473883)

[Hình 2.2.2‑1: Bước 1 cài đặt Composer 16](#_Toc139473884)

[Hình 2.2.2‑2: Bước 2 cài đặt Composer 16](#_Toc139473885)

[Hình 2.2.2‑3: Bước 4 cài đặt Composer 17](#_Toc139473886)

[Hình 2.2.2‑4: Bước 5 cài đặt Composer 17](#_Toc139473887)

[Hình 2.2.2‑5: Bước 6 cài đặt Composer 18](#_Toc139473888)

[Hình 2.2.2‑6: Bước 7 cài đặt Composer 18](#_Toc139473889)

[Hình 2.2.2‑7: Bước 8 cài đặt Composer 19](#_Toc139473890)

[Hình 2.2.2‑8: Bước 9 cài đặt Composer 19](#_Toc139473891)

[Hình 2.2.2‑9: Bước 10 cài đặt Composer 20](#_Toc139473892)

[Hình 2.2.2‑10: Bước 11 cài đặt Composer 20](#_Toc139473893)

[Hình 2.2.4‑1: Giao diện trang chủ VSC 21](#_Toc139473894)

[Hình 2.2.4‑2: Bước 2 cài đặt VSC 22](#_Toc139473895)

[Hình 2.2.4‑3: Bước 3 cài đặt VSC 22](#_Toc139473896)

[Hình 2.2.4‑4: Bước 4 cài đặt VSC 23](#_Toc139473897)

[Hình 2.2.4‑5: Bước 5 cài đặt VSC 23](#_Toc139473898)

[Hình 2.2.4‑6: Bước 6 cài đặt VSC 24](#_Toc139473899)

[Hình 2.2.4‑7: Các extension cho PHP trên VSC. 24](#_Toc139473900)

[Hình 2.2.4‑8: Bước 1 cài đặt PHP trên VSC. 25](#_Toc139473901)

[Hình 2.2.4‑9: Bước 2 cài đặt PHP trên VSC. 25](#_Toc139473902)

[Hình 2.2.5‑1: Bước 1 cài đặt Laravel. 26](#_Toc139473903)

[Hình 2.2.5‑2: Bước 2 cài đặt Laravel. 26](#_Toc139473904)

[Hình 2.2.5‑3: Bước 3 cài đặt Laravel. 27](#_Toc139473905)

[Hình 2.2.5‑4: Bước 4 cài đặt Laravel. 27](#_Toc139473906)

[Hình 2.2.5‑5: Bước 5 cài đặt Laravel. 27](#_Toc139473907)

[Hình 2.2.5‑6: Bước 6 cài đặt Laravel. 28](#_Toc139473908)

[Hình 2.2.5‑7: Mô hình MVC 28](#_Toc139473909)

[Hình 2.2.6‑1: Bước 1 cài đặt Version Python. 29](#_Toc139473910)

[Hình 2.2.6‑2: Bước 2 cài đặt Version Python. 30](#_Toc139473911)

[Hình 2.2.6‑3: Bước 3 cài đặt Version Python. 30](#_Toc139473912)

[Hình 2.2.6‑4: Bước 4 cài đặt Version Python. 31](#_Toc139473913)

[Hình 2.2.6‑5: Cài đặt thành công Python trên windows. 31](#_Toc139473914)

[Hình 2.2.6‑6: Hiển thị vị trí cài đặt Python. 32](#_Toc139473915)

[Hình 2.2.6‑7: Cài đặt Python trên VSC. 32](#_Toc139473916)

[Hình 2.2.6‑8: Cài đặt thành công thư viện facebook\_scraper. 33](#_Toc139473917)

[Hình 3.1.2‑1: Mối quan hệ giữa fanpages và posts 35](#_Toc139473918)

[Hình 3.1.2‑2: Mối quan hệ giữa posts và posts\_items. 36](#_Toc139473919)

[Hình 3.1.2‑3: Mối quan hệ giữa fanpages và thanhvien 36](#_Toc139473920)

[Hình 3.1.3‑1: Mô hình quan hệ dữ liệu. 39](#_Toc139473921)

[Hình 3.1.3‑1: Sơ đồ thuật toán đồng bộ dữ liệu từ group facebook về MySQL 40](#_Toc139473922)

[Hình 3.1.3‑2: Các bảng được tạo khi đoạn code được chạy. 42](#_Toc139473923)

[Hình 3.1.3‑3: Dữ liệu được đưa vào bảng Fanpages. 43](#_Toc139473924)

[Hình 3.1.3‑4: Kết quả được lưu vào bảng posts 43](#_Toc139473925)

[Hình 3.1.3‑5: Kết quả được lưu vào bảng posts\_items 43](#_Toc139473926)

[Hình 3.1.3‑6:Sơ đồ thuật toán cập nhập bài đăng mới theo ngày 44](#_Toc139473927)

[Hình 3.1.3‑7: Khi chưa chạy chương trình cập nhập theo ngày. 46](#_Toc139473928)

[Hình 3.1.3‑8: Kết quả sau khi chạy code cập nhập theo ngày. 46](#_Toc139473929)

[Hình 3.1.3‑9: Sơ đồ thuật toán cập nhập lại nội dung bài đăng 47](#_Toc139473930)

[Hình 3.1.3‑10: Nội dung ban đâu hình ảnh khi chưa chỉnh sửa trên facebook. 50](#_Toc139473931)

[Hình 3.1.3‑11: Nội dung ban đâu hình ảnh khi chưa chỉnh sửa trong database 50](#_Toc139473932)

[Hình 3.1.3‑12: Sau chi chỉnh sửa nội dung trên facebook. 51](#_Toc139473933)

[Hình 3.1.3‑13: Kết quả sau khi chạy chương trình 51](#_Toc139473934)

[Hình 3.1.3‑14: Sơ đồ thuật toán cập nhập lại URL hình ảnh bị lỗi 52](#_Toc139473935)

[Hình 3.1.3‑15: Khi URL hình ảnh bị lỗi. 53](#_Toc139473936)

[Hình 3.1.3‑16: Kết quả sau khi sửa lỗi hình ảnh. 54](#_Toc139473937)

[Hình 4.1.2‑1: Sơ đồ hoạt động của user khi truy cập website. 59](#_Toc139473938)

[Hình 4.2.1‑1: Header và slider trang chủ. 60](#_Toc139473939)

[Hình 4.2.1‑2: Tìm kiếm và bộ lọc thời gian. 60](#_Toc139473940)

[Hình 4.2.1‑3: Tin tức mới cập nhập. 60](#_Toc139473941)

[Hình 4.2.1‑4: sidebar 61](#_Toc139473942)

[Hình 4.2.1‑5: Footer chung của tất cả các trang 61](#_Toc139473943)

[Hình 4.2.2‑1: Trang chi tiết bài viết. 61](#_Toc139473944)

[Hình 4.2.3‑1: Phần giới thiệu CLB của trang giới thiệu. 62](#_Toc139473945)

[Hình 4.2.3‑2: Thành viên của CLB. 62](#_Toc139473946)

# DANH MỤC CÁC BẢNG

[Bảng 3.1‑1: Mô tả table Fanpages 34](#_Toc139369400)

[Bảng 3.1‑2: Mô tả table posts 34](#_Toc139369401)

[Bảng 3.1‑3: Table posts\_items 35](#_Toc139369402)

[Bảng 3.1‑4: Mô tả table Thanhvien 35](#_Toc139369403)

[Bảng 3.1‑5: Mô tả từ điển CSDL fanpages 36](#_Toc139369404)

[Bảng 3.1‑6: Mô tả từ điển CSDL posts 37](#_Toc139369405)

[Bảng 3.1‑7: Mô tả từ điển CSDL posts\_items 37](#_Toc139369406)

[Bảng 3.1‑8: Mô tả từ điển CSDL thanhvien 38](#_Toc139369407)

[Bảng 3.3‑1: Đồng bộ dữ liệu từ Group facebook và lưu về database 40](#_Toc139369408)

[Bảng 3.3‑2: Thực hiện việc cập nhập bài đăng mới theo ngày 44](#_Toc139369409)

[Bảng 3.3‑3: Thực hiện việc cập nhập lại nội dung bài đăng 47](#_Toc139369410)

[Bảng 3.3‑4: Thực hiện việc cập nhập lại URL hình ảnh khi bị lỗi 52](#_Toc139369411)

[Bảng 3.4‑1: postModel 54](#_Toc139369412)

[Bảng 3.4‑2: images\_Post\_Model 55](#_Toc139369413)

[Bảng 3.4‑3: ThanhVienModel 55](#_Toc139369414)

[Bảng 3.4‑4: caulacboModel 55](#_Toc139369415)

[Bảng 3.4‑5: CLB\_Model 56](#_Toc139369416)

[Bảng 3.4‑6: IndexController.php 56](#_Toc139369417)

[Bảng 3.4‑7: web.php 58](#_Toc139369418)

# 

# TỔNG QUAN VỀ ĐỀ TÀI

*Chúng em xây dựng một trang web quản lý thông tin cho các câu lạc bộ trên Facebook. Mục tiêu của chúng em là tạo ra một giao diện thân thiện, dễ sử dụng để quản lý thông tin câu lạc bộ. Chúng em sẽ sử dụng ngôn ngữ lập trình web như HTML, CSS, python và PHP kết hợp với các công cụ khác nhau. Đề tài này nhằm nâng cao hiệu quả quản lý thông tin câu lạc bộ trên Facebook và mang lại lợi ích cho thành viên và người quản lý.*

## Giới thiệu

Trong thời đại số hóa ngày nay, việc xây dựng và phát triển các trang web đóng vai trò quan trọng trong việc kết nối và chia sẻ thông tin trực tuyến. Trong tiểu luận này, chúng em giới thiệu một trang web đặc biệt được phát triển với mục tiêu hiển thị thông tin bài đăng và thông tin thành viên CLB.

Với sự sử dụng chủ đạo của ngôn ngữ lập trình Python, chúng em đã xây dựng một cơ sở dữ liệu MySQL để đồng bộ dữ liệu bài đăng và lưu trữ chúng. Bằng cách kết hợp HTML và CSS, chúng em đã thiết kế giao diện web đẹp mắt và thân thiện với người dùng, tạo nên một khung chức năng chuyên nghiệp và hấp dẫn.

Với sự hỗ trợ của framework Laravel, trang web của chúng em có khả năng xử lý thông tin từ cơ sở dữ liệu MySQL và hiển thị chúng một cách dễ dàng trên trình duyệt web. Điều này cho phép người dùng dễ dàng truy cập và tìm hiểu thông tin về các bài đăng và thành viên trong CLB.

Hơn nữa, chúng em đã tạo ra các chức năng tìm kiếm và lọc trên trang web, giúp người dùng dễ dàng tìm kiếm thông tin theo các tiêu chí cụ thể. Qua quá trình tối ưu hóa và kiểm tra, chúng em đã đảm bảo rằng trang web hoạt động ổn định và mang đến trải nghiệm tốt trên nhiều thiết bị và kích thước màn hình khác nhau.

## Lý do chọn đề tài

Chúng em chọn đề tài "xây dựng web quản lý thông tin cho các câu lạc bộ của trường học " vì nó mang lại nhiều lợi ích và giải quyết một số vấn đề quan trọng. Đầu tiên, trang web này sẽ giúp quản lý thông tin một cách dễ dàng hơn. Với việc có một nền tảng trực tuyến cho các CLB, có thể tập trung và tổ chức thông tin về hoạt động, thành viên, lịch trình và các tài liệu quan trọng khác một cách hiệu quả. Điều này giúp tiết kiệm thời gian và nỗ lực trong việc tìm kiếm và quản lý thông tin.

Thứ hai, trang web này sẽ cung cấp một nguồn thông tin hữu ích cho các bạn học sinh, sinh viên mới vào trường và các bậc phụ huynh. Họ có thể tìm hiểu về các CLB mà họ quan tâm và muốn tham gia. Thông qua trang web, các thông tin về mục tiêu, hoạt động, thành viên và những lợi ích mà CLB mang lại sẽ được truyền tải một cách rõ ràng và đáng tin cậy.

Thứ ba, trang web sẽ giúp các thầy cô giáo quản lý hoạt động của CLB một cách dễ dàng hơn. Thay vì phải tốn nhiều nguồn lực và công sức để nắm bắt thông tin và quản lý hoạt động của từng CLB, họ có thể sử dụng trang web để theo dõi và điều phối các hoạt động một cách hiệu quả. Điều này giúp giáo viên tập trung vào việc hướng dẫn và định hình chất lượng hoạt động của CLB.

Cuối cùng, trang web cũng sẽ cung cấp một nền tảng để quảng bá hình ảnh của CLB đến các nhà tài trợ. Nhờ vào các thông tin về hoạt động, thành tích và tầm ảnh hưởng của CLB mà được cập nhật trên trang web, những nhà tài trợ có thể dễ dàng hiểu rõ hơn về giá trị và tiềm năng của việc tài trợ cho CLB này.

Với những lợi ích này, việc tạo ra một trang web quản lý thông tin cho các câu lạc bộ là một ý tưởng đáng giá và đáp ứng được nhiều nhu cầu của các bên liên quan.

## Mục tiêu đề tài

Chúng em đã xây dựng một trang web đáp ứng các mục tiêu sau để tạo ra một nền tảng tìm kiếm thông tin các câu lạc bộ và quản lý hoạt động CLB của trường một cách dễ dàng và hiệu quả:

Trang web của chúng em được thiết kế để cung cấp một công cụ tìm kiếm mạnh mẽ, cho phép người dùng tìm kiếm thông tin về các CLB dựa trên nhiều tiêu chí khác nhau. Người dùng có thể tìm kiếm theo tên CLB, giới thiệu, mục tiêu, thành viên, hoạt động và sự kiện của CLB. Việc này giúp người dùng nhanh chóng tìm thấy các CLB phù hợp với sở thích và mục tiêu cá nhân.

Một trong những mục tiêu quan trọng khác của trang web là đảm bảo việc theo dõi và sao lưu bài đăng cụ thể của các CLB theo từng danh mục. Chúng em cung cấp một giao diện dễ sử dụng cho quản trị viên để lưu trữ và quản lý thông tin về các bài đăng, sự kiện và hoạt động của CLB. Quản trị viên có thể tạo danh mục theo thời gian, tên sự kiện hoặc các tiêu chí khác, giúp họ dễ dàng tra cứu và sao lưu thông tin theo yêu cầu.

Tóm lại, trang web mà chúng em xây dựng đáp ứng mục tiêu của việc tạo ra một nền tảng tìm kiếm thông tin CLB, dễ dàng theo dõi và sao lưu bài đăng theo danh mục, cũng như cung cấp một công cụ quản lý hiệu quả cho nhà trường. Chúng em tin rằng trang web sẽ tạo ra một môi trường tương tác thuận tiện và hỗ trợ sự phát triển của các CLB trong trường.

## Phương pháp nghiên cứu

Để thực hiện đề tài xây dựng trang web quản lý thông tin của các câu lạc bộ trên Facebook, chúng em đã sử dụng các công nghệ sau đây:

Sử dụng HTML và CSS: chúng em đã sử dụng HTML và CSS để xây dựng giao diện trang web. HTML được sử dụng để tạo cấu trúc và các thành phần của trang web, trong khi CSS được sử dụng để định dạng và trình bày các phần tử trên trang web, giúp tạo ra giao diện hấp dẫn và thân thiện với người dùng.

Sử dụng Python: Chúng em đã sử dụng ngôn ngữ lập trình Python để lấy dữ liệu bài đăng từ Facebook và lưu trữ vào cơ sở dữ liệu MySQL. Python cung cấp các thư viện và công cụ mạnh mẽ để tương tác với API của Facebook và truy xuất thông tin từ các câu lạc bộ trên Facebook. Dữ liệu bài đăng sau đó được lưu trữ vào cơ sở dữ liệu để có thể được sử dụng và hiển thị trên trang web.

Sử dụng framework Laravel của PHP: Chúng em đã sử dụng framework Laravel của PHP để thực hiện việc đưa dữ liệu từ cơ sở dữ liệu lên trang web. Laravel cung cấp một cấu trúc phát triển mạnh mẽ và dễ sử dụng, cho phép chúng em truy xuất dữ liệu từ cơ sở dữ liệu MySQL và hiển thị lên trang web một cách thuận tiện và linh hoạt. Nó cung cấp các tính năng như routing, ORM (Object-Relational Mapping), xử lý form và nhiều tính năng khác giúp tăng tính bảo mật và hiệu suất của trang web.

Nhờ sự kết hợp của HTML, CSS, Python và framework Laravel của PHP, chúng em đã xây dựng được trang web quản lý thông tin của các câu lạc bộ trên Facebook với giao diện hấp dẫn, dữ liệu được đồng bộ từ Facebook và hiển thị trên trang web một cách thuận tiện và dễ dàng quản lý.

## Kết luận:

Trong chương này, chúng em tóm tắt đề tài "Xây dựng web quản lý thông tin của group các Câu lạc bộ trên Facebook" với việc sử dụng HTML và CSS để xây dựng giao diện trang web. Python được sử dụng để lấy dữ liệu từ Facebook, tức là kết nối và truy vấn các thông tin cần thiết từ các group Câu lạc bộ trên nền tảng Facebook. Laravel, một framework PHP, được sử dụng để đưa dữ liệu từ cơ sở dữ liệu và hiển thị lên trang web. Điều này đảm bảo rằng thông tin quản lý của các Câu lạc bộ có thể được lưu trữ, truy xuất và hiển thị một cách hiệu quả trên trang web. Tóm lại, đề tài này kết hợp sự sáng tạo của HTML và CSS trong việc xây dựng giao diện, Python trong việc lấy dữ liệu từ Facebook và Laravel trong việc đưa dữ liệu từ cơ sở dữ liệu lên trang web, tạo nên một giải pháp toàn diện cho việc quản lý thông tin của các Câu lạc bộ trên Facebook.

# CƠ SỞ LÝ THUYẾT

*Trong chương này, chúng em sẽ giới thiệu sơ lược một số ngôn ngữ lập trình và công nghệ liên quan, bao gồm HTML, CSS, Python, PHP và framework Laravel của PHP. Chúng em sẽ phân tích các kỹ thuật được sử dụng trong đề tài bao gồm cách cài đặt môi trường cho mỗi ngôn ngữ lập trình, cách thiết lập các công cụ hỗ trợ lập trình như XAMPP, Composer và Visual Studio Code. Chúng em cũng sẽ giới thiệu cách tạo một dự án bằng Laravel và cách cài đặt các thư viện Python. Bên cạnh đó, chúng em sẽ có một cái nhìn tổng quan về MySQL và cách thực hiện các truy vấn cơ bản trên nền tảng này.*

## Giới thiệu

### . HTML

HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu được sử dụng để xây dựng và biểu diễn cấu trúc của các trang web. Nó sử dụng các thẻ để định dạng và xác định vai trò của các phần tử trên trang web. HTML cho phép tạo tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, biểu đồ, biểu mẫu và nhiều phần tử khác. Nó cung cấp cấu trúc và ý nghĩa cho các phần tử, giúp trình duyệt hiển thị nội dung đúng đắn. HTML là một phần quan trọng của việc phát triển trang web và là tiêu chuẩn được duy trì bởi W3C.

### . CSS

CSS (Cascading Style Sheets) là ngôn ngữ định dạng sử dụng để tạo kiểu cho các phần tử HTML trên trang web. Nó cho phép bạn điều chỉnh màu sắc, font chữ, kích thước, định vị và các thuộc tính khác của các phần tử. CSS giúp tách biệt nội dung và kiểu dáng, cho phép thay đổi giao diện mà không ảnh hưởng đến nội dung. Cú pháp CSS gọn gàng và sử dụng bộ chọn để xác định các phần tử cần áp dụng kiểu dáng.

### Python

Python là ngôn ngữ lập trình máy tính bậc cao, linh hoạt và dễ học. Nó được sử dụng rộng rãi trong việc xây dựng trang web, phần mềm, tự động hóa tác vụ và phân tích dữ liệu. Python có nhiều ưu điểm như đa nền tảng, dễ hiểu và hỗ trợ đa mô-đun. Nó được sử dụng trong nhiều lĩnh vực như phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa và đồ họa. Python có cộng đồng lập trình viên phát triển mạnh mẽ và tài liệu phong phú.

### Ngôn ngữ PHP

PHP là ngôn ngữ lập trình máy chủ được sử dụng để phát triển ứng dụng web. Nó là một ngôn ngữ mã nguồn mở, tương thích với nhiều nền tảng và dễ đọc. PHP cho phép kết nối với các thành phần HTML và thực hiện các chức năng như tạo máy chủ, xây dựng cơ sở dữ liệu, xử lý dữ liệu và tạo các ứng dụng phần mềm. Một số ứng dụng phổ biến của PHP bao gồm thiết lập chương trình máy chủ, tạo các dòng tập lệnh, xây dựng ứng dụng và hỗ trợ cơ sở dữ liệu.

### Framework Laravel

Laravel là một framework phát triển ứng dụng web mạnh mẽ và phổ biến được viết bằng ngôn ngữ PHP. Nó cung cấp cú pháp rõ ràng và đẹp, hệ thống routing, ORM tuyệt vời, bảo mật mạnh mẽ, quản lý giao diện người dùng và hỗ trợ cho việc kiểm thử. Laravel giúp đơn giản hóa và tăng tốc quá trình phát triển ứng dụng web, và được sử dụng rộng rãi trong ngành công nghiệp.

## Phân tích kỹ thuật

Phân tích kỹ thuật của đề tài "Xây dựng trang web quản lý thông tin của các câu lạc bộ trên Facebook" bao gồm :

* Giao diện trang web: Đề tài sử dụng HTML và CSS để xây dựng giao diện trang web. HTML được sử dụng để tạo cấu trúc và các thành phần của trang web, trong khi CSS được sử dụng để dịnh dạng và trình bày giao diện trực quan và hấp dẫn người dùng.
* Lấy dữ liệu từ Facebook: Để đồng bộ dữ liệu từ Facebook, đề tài sử dụng thư viện "facebook\_scraper" trong ngôn ngữ lập trình Python. Thư viện này cung cấp các chức năng để truy cập vào trang web Facebook và trích xuất thông tin từ các bài đăng và trang cá nhân của câu lạc bộ. Bằng cách sử dụng thư viện này, bạn có thể lấy dữ liệu bài đăng, thông tin thành viên, hoạt động và sự kiện của các câu lạc bộ mà không cần sử dụng trực tiếp API của Facebook.
* Lưu trữ dữ liệu: Dữ liệu bài đăng và thông tin câu lạc bộ từ Facebook được lưu trữ vào cơ sở dữ liệu MySQL. Python được sử dụng để kết nối và thao tác với cơ sở dữ liệu, đảm bảo việc lưu trữ và truy xuất dữ liệu một cách hiệu quả và bảo mật.
* Framework Laravel của PHP: Để xây dựng ứng dụng web, đề tài sử dụng framework Laravel của PHP. Laravel cung cấp một cấu trúc phát triển mạnh mẽ và dễ sử dụng, giúp quản lý routing, xử lý form, xác thực người dùng và tương tác với cơ sở dữ liệu MySQL. Framework này giúp tăng tính bảo mật, hiệu suất và khả năng mở rộng của trang web.

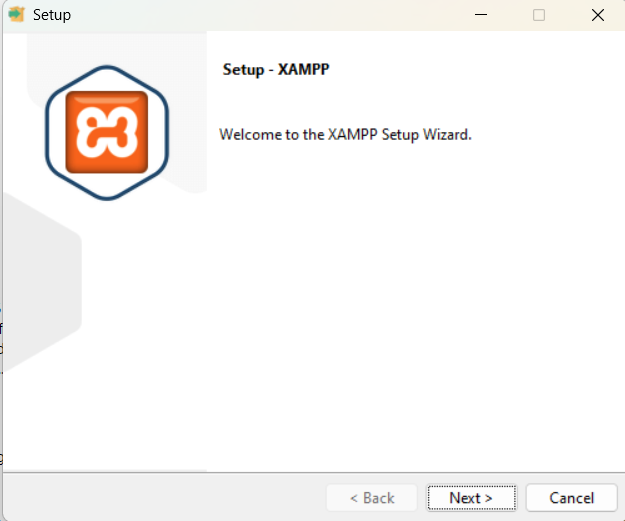
Tổng quan, đề tài sử dụng HTML, CSS, Python và framework Laravel của PHP để xây dựng giao diện trang web, lấy dữ liệu từ Facebook thông qua thư viện "facebook\_scraper", lưu trữ vào cơ sở dữ liệu MySQL và sử dụng framework Laravel để xây dựng ứng dụng web quản lý thông tin câu lạc bộ.

### Cài đặt Xampp

Phần mềm XAMPP là một loại ứng dụng phần mềm khá phổ biến và thường hay được các lập trình viên sử dụng để xây dựng và phát triển các dựa án website theo ngôn ngữ PHP. XAMPP được sử dụng cho mục đích nghiên cứu, phát triển website qua Localhost của máy tính cá nhân.

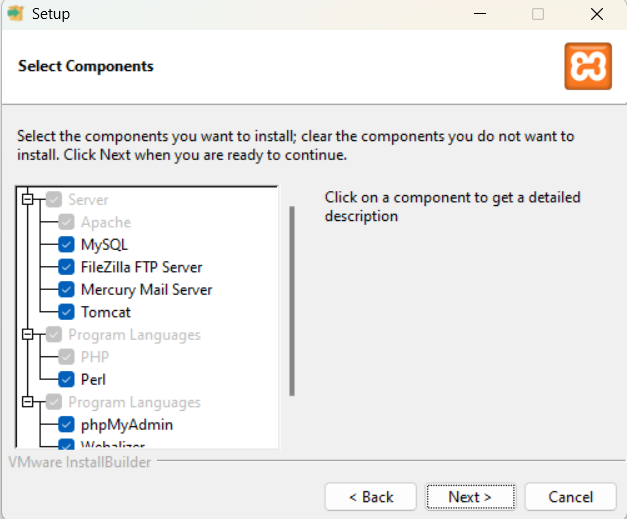
Đầu tiên phải tải Xampp từ trang chủ của Xampp sau đó mở file lên và tiến hành cài đặt.

* Bước 1: Tải xampp về và nhấn file .exe để cài đặt



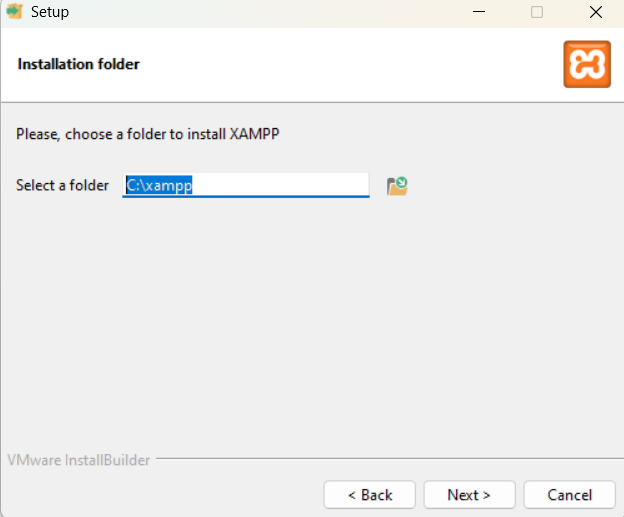
Hình 2.2.1‑1: Bước 1 cài đặt Xampp

* Bước 2: Nhấn next



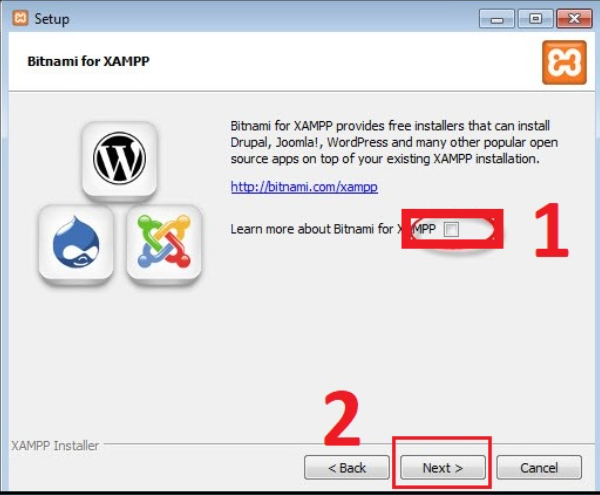
Hình 2.2.1‑2: Bước 2 cài đặt Xampp

* Bước 3: Chọn nơi lưu trữ phần mềm và Next.



Hình 2.2.1‑3: Bước 3 Cài đặt Xampp

* Bước 4: Bỏ chọn phần Learn more about Bitnami for XAMPP > Nhấn Next 2 lần nữa để bắt đầu quá trình cài đặt XAMPP.



Hình 2.2.1‑4: Bước 4 cài đặt Xampp

* Bước 5: Nhấn Finish để kết thúc cài đặt và mở bảng điều khiển của XAMPP.

### Cài đặt Composer

Composer là một công cụ quản lý phụ thuộc cho dự án PHP, giúp cài đặt và quản lý các thư viện và gói phụ thuộc một cách tự động và dễ dàng.

* Bước 1: Truy cập vào đường link : <https://getcomposer.org/>

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 2.2.2‑1: Bước 1 cài đặt Composer

* Bước 2: Nhập vào phần download

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Hình 2.2.2‑2: Bước 2 cài đặt Composer

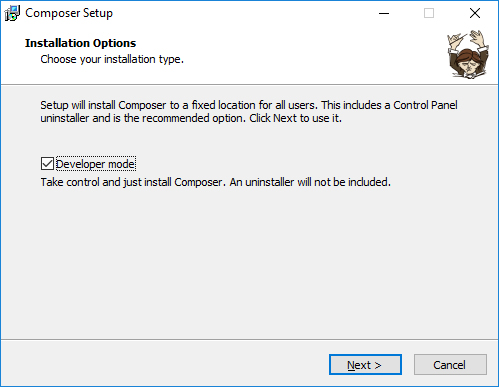
* Bước 3: Download file  [Composer-Setup.exe](https://getcomposer.org/Composer-Setup.exe)  về máy
* Bước 4: Trong thư mục lưu file [Composer-Setup.exe](https://getcomposer.org/Composer-Setup.exe) vừa được tải về. Double click vào file vừa download và tiến hành cài đặt.

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

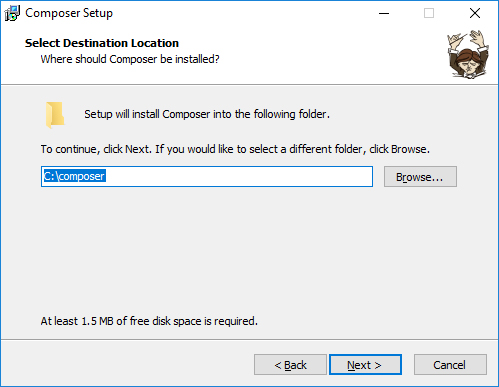
Hình 2.2.2‑3: Bước 4 cài đặt Composer

* Bước 5: Nhấn next để tiếp tục.



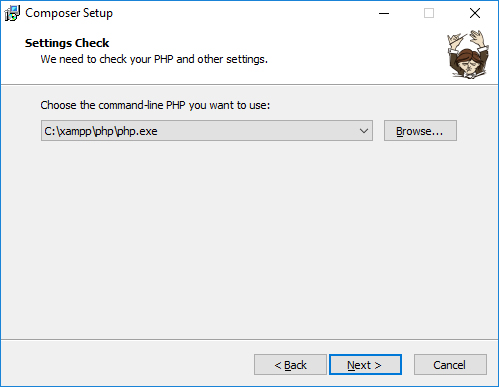
Hình 2.2.2‑4: Bước 5 cài đặt Composer

* Bước 6: Chọn nơi lưu trữ và nhấn next



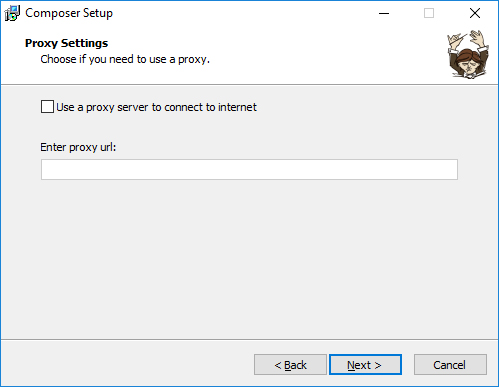
Hình 2.2.2‑5: Bước 6 cài đặt Composer

* Bước 7: Chọn nơi lưu trữ file PHP và bấm Next



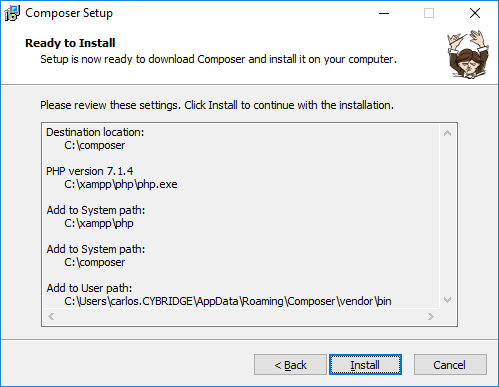
Hình 2.2.2‑6: Bước 7 cài đặt Composer

* Bước 8:Bấm Next để tiếp tục



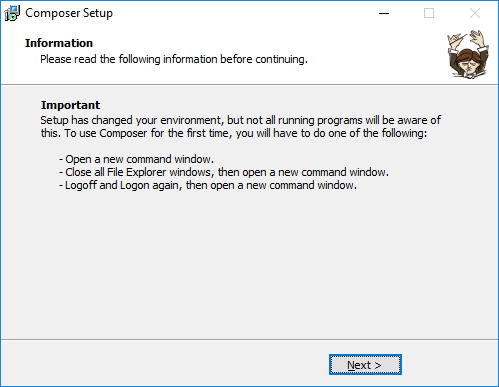
Hình 2.2.2‑7: Bước 8 cài đặt Composer

* Bước 9:Bấm Install để tiến hành cài đặt



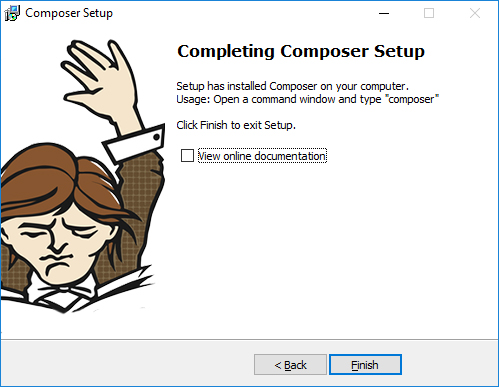
Hình 2.2.2‑8: Bước 9 cài đặt Composer

* Bước 10: Bảng thông tin hướng dẫn, Click "Next".



Hình 2.2.2‑9: Bước 10 cài đặt Composer

* Bước 11: Chọn Finish để hoàn tất cài đặt



Hình 2.2.2‑10: Bước 11 cài đặt Composer

### Lập trình cơ sở dữ liệu MySQL

#### Kết nối cơ sở dữ liệu

Hàm mysql\_connect() sẽ kết nối tới MySQL server. Ta có:

mysql\_connect( $host, $username, $pass)

Trong đó:

* $host là tên hosting.
* $username là tên người dùng MySQL.
* $pass là mật khẩu của người dùng.

#### Thực thi các câu lệnh truy vấn

Mysqli\_query là một hàm trong PHP dùng để thực hiện một truy vấn SQL tới cơ sở dữ liệu MySQL. Hàm này cung cấp cho chúng ta cách tiếp cận đến cơ sở dữ liệu và thực hiện các truy vấn SELECT, INSERT, UPDATE và DELETE trên cơ sở dữ liệu. Kết quả trả về từ hàm mysqli\_query có thể là một đối tượng mysqli\_result hoặc TRUE nếu truy vấn thành công hoặc FALSE nếu truy vấn thất bại.

Ví dụ:

$conn = mysqli\_connect("host", "username", "password", "database\_name");

$query = "SELECT \* FROM users";

$result = mysqli\_query($conn, $query);

Sau khi gọi hàm mysqli\_query, kết quả trả về sẽ là một đối tượng kết quả hoặc false nếu truy vấn không thành công. Đối tượng kết quả này có thể sử dụng để lấy dữ liệu từ cơ sở dữ liệu.

### Cài đặt PHP

Muốn sử dụng ngôn ngữ PHP đầu tiên cần 1 công cụ để lập trình. Ở trong đề tài này chúng em lựa chọn công cụ Visual Studio Code.

* Cài đặt Visual Studio Code

Đầu tiên truy cập vào <https://code.visualstudio.com/> để download phần mềm visual studio code.

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Biểu tượng máy tính

Mô tả được tạo tự động

Hình 2.2.4‑1: Giao diện trang chủ VSC

* Bước 2: Chọn vào phần Download for Windows. File cài đặt sẽ được lưu về máy.

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 2.2.4‑2: Bước 2 cài đặt VSC

* Bước 3: Sau khi tải xong thì chạy file VSCodeUserSetup.exe, chọn I accept the agreement và bấm Next để tiếp tục

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Hình 2.2.4‑3: Bước 3 cài đặt VSC

* Bước 4: Chọn Create a desktop icon để tạo đường dẫn tắt ở cửa sổ window ( cũng có thể không chọn ) và bấm Next.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

Hình 2.2.4‑4: Bước 4 cài đặt VSC

* Bước 5: Chọn Install để tiến hành cài đặt.

*Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động*

Hình 2.2.4‑5: Bước 5 cài đặt VSC

* Bước 6:Chọn Lunch VSC và Finish để kết thúc việc cài đặt và khởi chạy phần mềm.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự động

Hình 2.2.4‑6: Bước 6 cài đặt VSC

* Cài đặt PHP trên VSC

Sau khi cài đặt thành công phần mềm Visual Studio Code, tiến hành cài đặt PHP trên VCS để phục vụ cho đề tài.

Đầu tiên để bắt đầu lập trình PHP sử dụng VSC đầu tiên phải cài 1 số Extension hỗ trợ. Các extension cho PHP có rất nhiều trên VS Code Marketplace nhưng dưới đây là những VS Code Extension được khuyến khích sử dụng:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự động

Hình 2.2.4‑7: Các extension cho PHP trên VSC.

Trong đó:

* [PHP IntelliSense](https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-intellisense): Gợi nhớ code (Suggest), Tự động hoàn thành code (Auto completion)
* [Code Runner](https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner): Chạy PHP Script
* [PHP Debug](https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-debug): Dành cho việc debug ứng dụng PHP, script PHP
* [HTML CSS Support](https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css): Hỗ trợ viết mã HTML/CSS

Để cài đặt PHP trên VSC thực hiện các bước sau:

* Bước 1: bấm tổ hợp phím [ Ctrl + Shift + X] hoặc bấm vào biểu tượng Extension trên VS Code, tìm kiếm extension cần thiết và bấm [ Install ] để cài đặt.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 2.2.4‑8: Bước 1 cài đặt PHP trên VSC.

* Bước 2: Từ cửa sổ của Visual Code -> [ **File** ] -> [ **Preferences** ] -> [ **Settings** ] (Hoặc sử dụng tổ hợp phím [ **Ctrl +,**]) để mở User Settings.

Ở cửa sổ [ **User Settings** ] thêm hoặc chỉnh sửa các thông tin sau đây:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình 2.2.4‑9: Bước 2 cài đặt PHP trên VSC.

**Trong đó:**

* **php.suggest.basic: false** -> Vô hiệu quả chức năng nhắc code có sẵn của VS Code, để chỉ sử dụng chức năng của Extension, tránh bị dupplicate kết quả khi gõ code trên Editor.
* **php.validate.enable: true** -> Sử dụng chức năng validate source code (Sử dụng php linter)
* **php.validate.executablePath** -> Đường dẫn tới file thưc thi PHP, cho tác vụ validate
* **php.validate.run**: “onSave” -> Thực hiện validate khi save file
* **php.executablePath** -> Đường dẫn tới file thực thi PHP, cho tác vụ run, debug,…

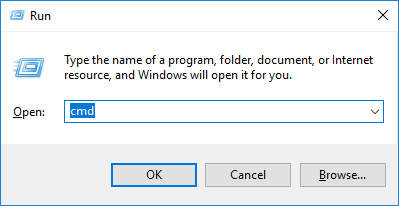
### Cài đặt Framework Laravel và mô hình MVC

#### Cài đặt Framework Laravel

Laravel được phát triển bởi Taylor Otwell với phiên bản cho ra mắt đầu tiên là vào tháng 6 năm 2011. Đây là một framework có mã nguồn mở miễn phí cùng với cú pháp mạch lạc, rõ ràng. Laravel ra đời nhằm mục đích hỗ trợ cho các ứng dụng web, lấy cốt lõi dựa theo mô hình MVC (Model – View – Controller).

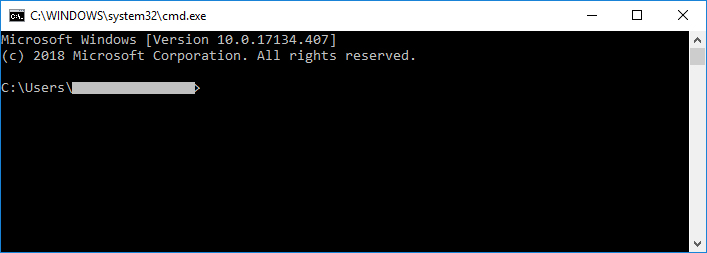
Tạo project laravel thông qua Composer:

* + Bước 1: Thao tác lệnh Composer bằng cmd, mở cửa sổ cmd bằng cách nhấn tổ hợp phìm "window + R", gõ "cmd", nhấn "Ok" để mở cửa số cmd Window.



Hình 2.2.5‑1: Bước 1 cài đặt Laravel.

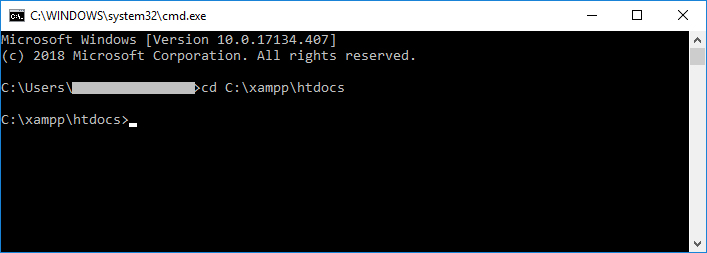
* + Bước 2: Cửa sổ làm việc cmd:



Hình 2.2.5‑2: Bước 2 cài đặt Laravel.

* Bước 3: Mặc định của Xampp để chạy được project PHP là cần đặt trong thư mục *C:\xampp\htdocs*, do đó cần thao tác trong thư mục này, để trỏ tới thư mục này gõ lệnh sau và enter:

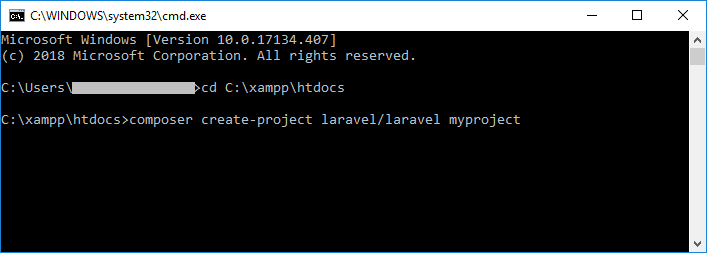
cd C:\xampp\htdocs



Hình 2.2.5‑3: Bước 3 cài đặt Laravel.

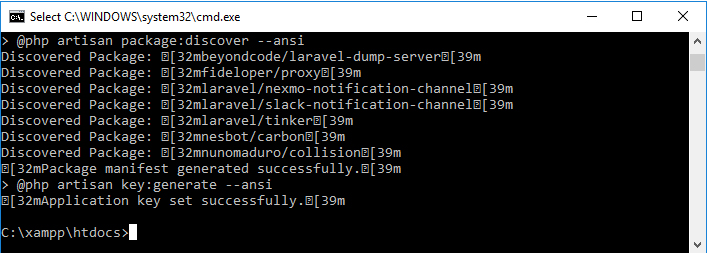
* Bước 4: Tạo project Laravel với tên là myproject bằng lệnh:

composer create-project laravel/laravel myproject



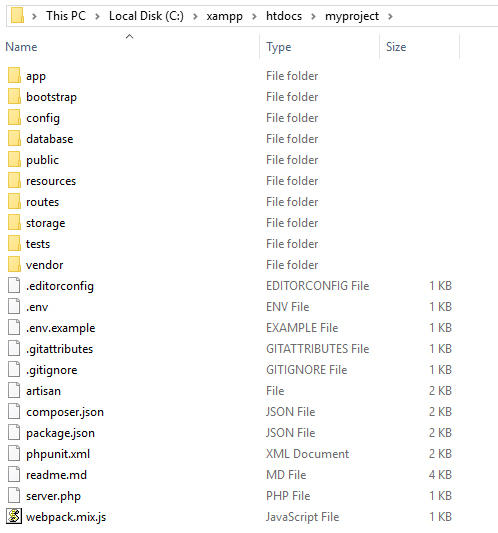
Hình 2.2.5‑4: Bước 4 cài đặt Laravel.

* Bước 5: Màn hình sau đây là kết quả tạo project Laravel thành công:



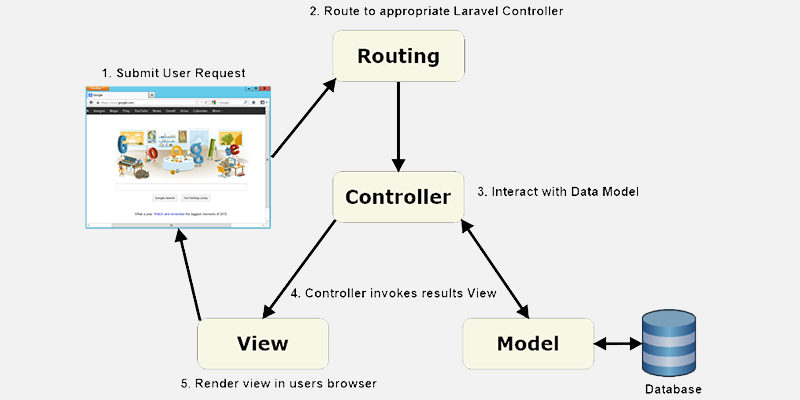
Hình 2.2.5‑5: Bước 5 cài đặt Laravel.

* Bước 6: Kiểm tra trong thư mục *C:\xampp\htdocs* xem có "myproject" được tạo chưa, nếu mở thư mục "myproject" mà có các file sau đây là thành công:



Hình 2.2.5‑6: Bước 6 cài đặt Laravel.

#### Mô hình MVC



Hình 2.2.5‑7: Mô hình MVC

MVC là viết tắt của ba từ Model – View – Controller. Mọi Request từ phía người dùng đều phải qua Route, dữ liệu được gửi xuống Controller để xử lý, cần dữ liệu sẽ lấy từ Model lên hoặc cập nhật dữ liệu xuống Model, kết quả gửi ra View cho người sử dụng. Trong đó:

* Model: cấu trúc dữ liệu theo cách tin cậy và chuẩn bị dữ liệu theo lệnh của controller. Tạo model bằng câu lệnh:

php artisan make:model PostModel

* View: Hiển thị dữ liệu cho người dùng theo cách dễ hiểu dựa trên hành động của người dùng.
* Controller Nhận lệnh từ người dùng, gửi lệnh đến cho Model để cập nhập dữ liệu, truyền lệnh đến View để cập nhập giao diện hiển thị. Tạo model bằng câu lệnh:

php artisan make:controller IndexController

### Cài đặt Python và thư viện facebook\_scraper.

* Cài đặt Version Python

Các phiên bản (version) của Python có 2 nhánh chính là Python 2 và Python 3. Chúng khá là khác nhau. Python 2 được ra mắt vào năm 2000 và phát triển đến version 2.7.18 thì đã ngưng phát triển. Python 3 được ra mắt vào năm 2008 và không hoàn toàn tương thích với Python 2. Python 3 vẫn đang được phát triển và đến nay (15/06/2023) thì đã có phiên bản Python 3.11.4.

* Bước 1: Đầu tiên truy cập vào trang: <https://www.python.org/> để tải xuống version Python phù hợp.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 2.2.6‑1: Bước 1 cài đặt Version Python.

* Bước 2: Sau đó nhấp vào phần Downloads để tải phiên bản Python phù hợp.

Ở đây nhấp chuột vào phần Downloads for Windows để tải xuống Python version 3.11.4 về máy tính.

Ảnh có chứa văn bản, phần mềm, Trang web, Website

Mô tả được tạo tự động

Hình 2.2.6‑2: Bước 2 cài đặt Version Python.

* Bước 3: Ở file vừa tải xuống double click vào file cài đặt Python để bắt đầu cài đặt và tick vào phần use admin và Add python.exe to PATH và click vào install Now để tiến hành cài đặt.

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Hình 2.2.6‑3: Bước 3 cài đặt Version Python.

* Bước 4: Bấm Close để hoàn thành tất tiến hành cài đặt.

*Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Mô tả được tạo tự động*

Hình 2.2.6‑4: Bước 4 cài đặt Version Python.

Sau khi cài đặt xong có thể tiến hành kiểm tra xem Python đã có trên Windows hay chưa, trước tiên mở ứng dụng dòng lệnh, chẳng hạn như PowerShell (Các bạn chỉ cần nhấn ô cửa sổ trên bàn phím, gõ chữ "Powershell" và nhấn Enter) hoặc mở cửa sổ dòng lệnh cmd.

Sau khi cửa sổ dòng lệnh được mở hãy nhập câu lệnh sau và nhấn nút Enter:

|  |
| --- |
| python --version |

Bằng cách sử dụng tùy chọn --version, phiên bản của Python đã được cài đặt sẽ được hiển thị trên màn hình. Ngoài ra, bạn có thể sử dụng tùy chọn -V:

|  |
| --- |
| python -V |

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình 2.2.6‑5: Cài đặt thành công Python trên windows.

Nếu muốn biết vị trí cài đặt, thì bạn có thể sử dụng lệnh where.exe trong cmd.exe hoặc PowerShell:

|  |
| --- |
| where.exe python |

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

Mô tả được tạo tự động

Hình 2.2.6‑6: Hiển thị vị trí cài đặt Python.

* Cài đặt Python trên VSC.

Tại giao diện Visual Studio Code, nhấn tổ hợp phím **Ctrl + Shift + X** -> nhập từ khóa "Python" chọn python extension -> click install:

*Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động*

Hình 2.2.6‑7: Cài đặt Python trên VSC.

* Cài đặt thư viên facebook\_scraper

Facebook\_Scraper Python là một thư viện Python mạnh mẽ để lấy dữ liệu từ Facebook. Thư viện này cung cấp các công cụ và API cho việc trích xuất thông tin từ trang cá nhân, trang doanh nghiệp, nhóm, bài viết và các tài khoản khác trên Facebook.

Đặc điểm chính:

* Trích xuất thông tin trang cá nhân, trang doanh nghiệp, nhóm, bài viết và các tài khoản khác trên Facebook.
* Lấy dữ liệu về bài viết, bình luận, lượt thích, chia sẻ và các hoạt động khác.
* Hỗ trợ việc đăng nhập vào Facebook để truy cập dữ liệu không công khai (public) hoặc dữ liệu có giới hạn (restricted).

Để có thể cài đặt được thư viện của python thì hệ thống cần phải đáp ứng 1 số các yêu cầu sau:

* Python 3.x
* Cài đặt các gói phụ thuộc: pip install -r requirements.txt

Ta có thể cài đặt thư viện facebook\_scraper bằng cách dùng Teminal của VSC hoặc cmd bằng cách nhập lệnh:

|  |
| --- |
| pip install facebook-scraper |

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, đen và trắng

Mô tả được tạo tự động

Hình 2.2.6‑8: Cài đặt thành công thư viện facebook\_scraper.

## Kết luận

Trong chương này, chúng em đã giới thiệu một bộ công cụ mạnh mẽ để xây dựng trang web quản lý thông tin của các Câu lạc bộ trên Facebook. Sử dụng HTML và CSS, chúng em đã tạo giao diện hấp dẫn và thu hút người dùng. Bằng cách sử dụng Python và thư viện facebook\_scraper, chúng em đã thu thập dữ liệu từ các group Câu lạc bộ trên Facebook. Với Laravel, một framework PHP đáng tin cậy, chúng em đã xây dựng và quản lý cơ sở dữ liệu một cách hiệu quả.

Tiếp theo, chúng em đã hướng dẫn cài đặt các công cụ cần thiết. Đầu tiên, XAMPP đã được cài đặt để tạo môi trường phát triển web local. Tiếp theo, sử dụng Composer, chúng em đã cài đặt Laravel và quản lý các phụ thuộc của nó. Visual Studio Code đã được sử dụng làm trình soạn thảo mã nguồn hiệu quả.

Sau đó, chúng em đã hướng dẫn cài đặt thư viện facebook\_scraper thông qua pip, trình quản lý gói Python.

Tóm lại, chương này đã giới thiệu các công cụ và ngôn ngữ quan trọng để xây dựng trang web quản lý thông tin của các Câu lạc bộ trên Facebook. Sử dụng HTML, CSS, Python và Laravel, chúng em đã tạo ra một ứng dụng mạnh mẽ và chuyên nghiệp.

# PHÂN TÍCH HỆ THỐNG

*Chúng em sẽ thực hiện việc phân tích và thiết kế database cho đề tài, phân tích các mối quan hệ trong database, xây dựng sơ đồ thuật toán và đồng bộ dữ liệu từ group facebook bằng thư viện facebook\_scraper và lưu về database. Dùng Laravel để lấy dữ liệu từ database và hiển thị lên website.*

## Phân tích mối quan niệm dữ liệu:

### Mô tả thực thể kết hợp diễn đạt các thành phần dữ liệu ở mức quan niệm dựa trên bảng danh mục các dữ liệu cơ bản.

Các loại thực thể có những thuộc tính sau:

* **fanpages**

Bảng 3.1‑1: Mô tả table Fanpages

|  |  |
| --- | --- |
| Thuộc tính | Tân từ |
| id  name  url  HinhAnh  gioithieu  ten | Mỗi câu lạc bộ có 1 id duy nhất để phân biệt với các câu lạc bộ khác, id tự tăng. Mỗi id còn được xác định bởi tên viết tắt câu lạc bộ (name), đường dẫn câu lạc bộ (url), hình ảnh của câu lạc bộ (HinhAnh), giới thiệu câu lạc bộ (gioithieu), tên đầy đủ của câu lạc bộ (ten) |

* Posts:

Bảng 3.1‑2: Mô tả table posts

|  |  |
| --- | --- |
| Thuộc tính | Tân từ |
| id  fanpage\_name  post\_id  user\_name  post\_content  created\_time  image  fbid\_image | Mỗi bài đăng có 1 id duy nhất để phân biệt với các câu lạc bộ khác, id tự tăng. Mỗi bài viết còn được xác định bởi bài viết của câu lạc bộ nào (fanpage\_name), người đăng bài (user\_name), nội dung bài viết (post\_content), ngày đăng (created\_time), hình ảnh bài viết (image) và đường dẫn đuôi của link hình ảnh (fbid\_image). |

* Posts\_item:

Bảng 3.1‑3: Table posts\_items

|  |  |
| --- | --- |
| Thuộc tính | Tân từ |
| id  fanpage\_name  post\_id  image\_items  image\_content  fbid\_image\_items | Mỗi bài đăng có thể có nhiều hình ảnh. Các hình ảnh trong 1 bài đăng sẽ được cập nhật vô posts\_items. Mỗi hình ảnh sẽ được đánh số thứ tự tự động (id), có tên fanpage của bài post đó (fanpage\_name), id bài viết (post\_id ), đường dẫn của từng hình ảnh (image\_items), nội dung hình ảnh (image\_content) và fbid của hình ảnh đó(fbid\_image\_items) |

* thanhvien

Bảng 3.1‑4: Mô tả table Thanhvien

|  |  |
| --- | --- |
| Thuộc tính | Tân từ |
| id  HinhThe  HovaTen  Khoa  NamHoc  ChucVu  fanpage\_id | Mỗi thành viên trong câu lạc bộ được đánh số thứ tự tự động (id) có hình thẻ (HinhThe), họ và tên (HovaTen), khoa (Khoa), năm học (NamHoc), chức vụ (ChucVu) và một khóa ngoại biểu thị câu lạc bộ trực thuộc (fanpage\_id). |

### Mô tả các mối kết hợp

* **fanpages và posts ( 1 - nhiều)**

**Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động**

Hình 3.1.2‑1: Mối quan hệ giữa fanpages và posts

* + 1 fanpages có thể có nhiều bài posts
  + 1 bài post chỉ nằm trong 1 fanpage
* posts và posts\_items ( 1 – nhiều )

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình 3.1.2‑2: Mối quan hệ giữa posts và posts\_items.

* 1 bài posts có thể có nhiều hình ảnh, hoặc 1 hình ảnh.
* Nhiều hình ảnh cùng 1 bài posts.
* **fanpages và thanhvien ( 1 – nhiều )**

A picture containing text, screenshot, diagram, font

Description automatically generated

Hình 3.1.2‑3: Mối quan hệ giữa fanpages và thanhvien

* 1 câu lạc bộ có nhiều thành viên.
* Nhiều thành viên có trong câu lạc bộ.

### Từ điển cơ sở dữ liệu

* **Câu lạc bộ**

Bảng 3.1‑5: Mô tả từ điển CSDL fanpages

|  |  |  |
| --- | --- | --- |
| fanpages | | |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| id | int(255) | Khóa chính, tự động tăng |
| name | varchar(255) | Tên viết tắt câu lạc bộ |
| url | varchar(255) | Đường dẫn link group |
| HinhAnh | varchar(1000) | Hình đại điện group |
| gioithieu | text | Đoạn văn giới thiệu group |
| Ten | text | Tên câu lạc bộ |

* **Bài viết**

Bảng 3.1‑6: Mô tả từ điển CSDL posts

|  |  |  |
| --- | --- | --- |
| posts | | |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| id | int(255) | Khóa chính, tự động tăng |
| fanpage\_name | varchar(255) | Tên viết tắt câu lạc bộ |
| post\_id | varchar(255) | Id bài viết |
| user\_name | varchar(255) | Người đăng bài post |
| post\_content | text | Nội dung bài post |
| created\_time | datetime | Ngày đăng bài |
| image | text | Hình ảnh bài viết |
| fbid\_image | varchar(255) | fbid của hình ảnh |

* **Các hình ảnh trong bài đăng:**

Bảng 3.1‑7: Mô tả từ điển CSDL posts\_items

|  |  |  |
| --- | --- | --- |
| posts\_items | | |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| id | int(255) | Khóa chính, tự động tăng |
| fanpage\_name | varchar(255) | Tên viết tắt câu lạc bộ |
| post\_id | varchar(255) | Id hình ảnh |
| image\_items | text | Hình ảnh bài biết |
| image\_content | text | Nội dung bài post |
| fbid\_image\_items | varchar(255) | fbid của hình ảnh |

* **Thành viên câu lạc bộ**

Bảng 3.1‑8: Mô tả từ điển CSDL thanhvien

|  |  |  |
| --- | --- | --- |
| thanhvien | | |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| id | int(255) | Khóa chính, tự động tăng |
| HinhThe | text | Hình đại diện thành viên |
| HovaTen | varchar(1000) | Họ và tên thành viên |
| Khoa | text | Thành viên thuộc khoa nào |
| NamHoc | datetime | Thành viên nhập học năm nào |
| ChucVu | text | Chức vụ thành viên |
| fanpage\_id | varchar(1000) | Dùng để so 1-nhiều với id trong field fanpages |

## Mô hình quan hệ (Relational Data Diagram )

A screenshot of a computer

Description automatically generated with low confidence

Hình 3.1.3‑1: Mô hình quan hệ dữ liệu.

## Dùng thư viện facebook\_scraper để đồng bộ dữ liệu từ group về MySQL

A screenshot of a cell phone

Description automatically generated

Hình 3.1.3‑1: Sơ đồ thuật toán đồng bộ dữ liệu từ group facebook về MySQL

Bảng 3.3‑1: Đồng bộ dữ liệu từ Group facebook và lưu về database

|  |
| --- |
| import mysql.connector  from facebook\_scraper import get\_posts  from urllib.parse import urlparse, parse\_qs  import requests  # Kết nối tới cơ sở dữ liệu MySQL  mydb = mysql.connector.connect(      host="localhost",      user="root",      password="",      database="data\_clb"  )  mycursor = mydb.cursor()  # Tạo bảng fanpages nếu chưa tồn tại  mycursor.execute("""CREATE TABLE IF NOT EXISTS fanpages (id INT(11) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,name VARCHAR(255) NOT NULL,url VARCHAR(255) NOT NULL,HinhAnh VARCHAR(1000) NOT NULL,gioithieu TEXT,Ten TEXT)""")  # Tạo bảng `posts` nếu chưa tồn tại  mycursor.execute("""CREATE TABLE IF NOT EXISTS posts (id INT AUTO\_INCREMENT PRIMARY KEY,fanpage\_name VARCHAR(255),post\_id VARCHAR(255) UNIQUE,user\_name VARCHAR(255),post\_content TEXT,created\_time DATETIME,image TEXT,fbid\_image VARCHAR(255))""")  # Tạo bảng `posts\_items` nếu chưa tồn tại  mycursor.execute("""CREATE TABLE IF NOT EXISTS posts\_items (id INT AUTO\_INCREMENT PRIMARY KEY,fanpage\_name VARCHAR(255),post\_id VARCHAR(255),image\_items TEXT,image\_content TEXT,fbid\_image\_items VARCHAR(255))""")  # Tạo bảng thanhvien nếu chưa tồn tại  mycursor.execute("""CREATE TABLE IF NOT EXISTS thanhvien (id INT(11) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,HinhThe VARCHAR(1000) NOT NULL,HovaTen TEXT NOT NULL,Khoa TEXT NOT NULL,NamHoc TEXT NOT NULL,ChucVu TEXT NOT NULL,clb\_id INT(255) NOT NULL)""")  # Thêm danh sách các fanpage vào bảng fanpages  fanpages = [      ('hcmusec', 'https://www.facebook.com/hcmusec'),      ('escussh', 'https://www.facebook.com/groups/escussh'),      ('valorantvietnam', 'https://www.facebook.com/groups/valorantvietnam')  ]  for fanpage in fanpages:      sql = "INSERT INTO fanpages (name, url) VALUES (%s, %s)"      val = (fanpage[0], fanpage[1])      mycursor.execute(sql, val)      mydb.commit()  # Lấy danh sách các fanpage từ bảng fanpages  mycursor.execute("SELECT \* FROM fanpages")  fanpages = mycursor.fetchall()  # Lấy dữ liệu bài đăng trên Facebook và đồng bộ vào cơ sở dữ liệu MySQL  for fanpage in fanpages:      for post in get\_posts(group=fanpage[1], pages=10):          # Kiểm tra xem post\_id đã tồn tại trong cơ sở dữ liệu chưa          mycursor.execute("SELECT \* FROM posts WHERE post\_id = %s", (post['post\_id'],))          result = mycursor.fetchone()          if result is None:              # Thêm dữ liệu vào bảng `posts`              sql = "INSERT INTO posts (fanpage\_name, post\_id, user\_name, post\_content, created\_time, image) VALUES (%s, %s, %s, %s, %s, %s)"              val = (fanpage[1], post['post\_id'], post['username'], post['text'], post['time'], post['image'])              mycursor.execute(sql, val)              mydb.commit()              # Trích xuất ID hình ảnh và cập nhật vào cột fbid\_image (nếu có)              if post['image'] is not None:                  image\_id = None                  parsed\_url = urlparse(post['image'])                  parsed\_qs = parse\_qs(parsed\_url.query)                  if 'fbid' in parsed\_qs:                      image\_id = parsed\_qs['fbid'][0]                  sql\_update = "UPDATE posts SET fbid\_image = %s WHERE post\_id = %s"                  val\_update = (image\_id, post['post\_id'])                  mycursor.execute(sql\_update, val\_update)                  mydb.commit()              # Thêm dữ liệu vào bảng `posts\_items` cho các hình ảnh khác (nếu có)              if post['images'] is not None and len(post['images']) > 1:                  for i in range(1, len(post['images'])):                      sql = "INSERT INTO posts\_items (fanpage\_name, post\_id, image\_items, image\_content) VALUES (%s, %s, %s, %s)"                      val = (fanpage[1], post['post\_id'], post['images'][i], post['text'])                      mycursor.execute(sql, val)                      mydb.commit()                      # Trích xuất ID hình ảnh và cập nhật vào cột fbid\_image\_items (nếu có)                      if post['images'][i] is not None:                          image\_id\_items = None                          parsed\_url\_items = urlparse(post['images'][i])                          parsed\_qs\_items = parse\_qs(parsed\_url\_items.query)                          if 'fbid' in parsed\_qs\_items:                              image\_id\_items = parsed\_qs\_items['fbid'][0]                          sql\_update\_items = "UPDATE posts\_items SET fbid\_image\_items = %s WHERE image\_items = %s"                          val\_update\_items = (image\_id\_items, post['images'][i])                          mycursor.execute(sql\_update\_items, val\_update\_items)                          mydb.commit()      # Lấy danh sách các bài đăng và cập nhật nội dung hình ảnh vào cơ sở dữ liệu      mycursor.execute("SELECT post\_id, fbid\_image\_items FROM posts\_items")      posts\_items = mycursor.fetchall()      for post\_id, fbid\_image\_items in posts\_items:          if fbid\_image\_items and post\_id:              url = "https://m.facebook.com/story.php?story\_fbid={fbid}&id={user\_id}".format(                  fbid=fbid\_image\_items,                  user\_id=post\_id              )              response = requests.get(url)              if response.status\_code == 200:                  posts = list(get\_posts(post\_urls=[url], cookies="cookies.txt"))                  if len(posts) > 0:                      image\_content = posts[0]["post\_text"]                      # Cập nhật dữ liệu vào cơ sở dữ liệu                      sql\_update\_items = "UPDATE posts\_items SET image\_content = %s WHERE post\_id = %s"                      val\_update\_items = (image\_content, post\_id)                      mycursor.execute(sql\_update\_items, val\_update\_items)                      mydb.commit() |

**Kết quả:**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động**

Hình 3.1.3‑2: Các bảng được tạo khi đoạn code được chạy.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

Hình 3.1.3‑3: Dữ liệu được đưa vào bảng Fanpages.

**Hình 3.3.2:** Các thông tin như link , name của group ta sẽ lưu vào bảng fanpages và các trường như HinhAnh, gioithieu, Ten ta sẽ không lưu thông tin vào và ta sẽ nhập thủ công để khi ta có muốn thay đổi tùy ý theo ý muốn của mình.

**Ảnh có chứa văn bản, số, Phông chữ, hàng

Mô tả được tạo tự động**

Hình 3.1.3‑4: Kết quả được lưu vào bảng posts

**Hình 3.3.3:** Cho ta thấy kết quả sau khi ta chạy đoạn code để lấy dữ liệu của một bài đăng và lưu các thông tin của bài đăng đó vào bảng posts. Các thông tin được lưu là : id bài viết, bài đăng trên thuộc của group nào , tên tác giả bài đăng, nội dung bài đăng, thời gian đăng bài, đường dẫn hình ảnh của bài đăng, fbid của hình ảnh bài đăng nếu bài đăng đó có định dạng hình ảnh là : m.facebook .

**Ảnh có chứa văn bản, số, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động**

Hình 3.1.3‑5: Kết quả được lưu vào bảng posts\_items

**Hình 3.3.4:** Các dữ liệu lưu vào bảng posts\_item cũng tương tự như dữ liệu được lưu vào bảng posts nhưng khác ở chỗ. Nếu trong một bài đăng có nhiều hình ảnh và mỗi hình ảnh thì có nội dung bài đăng khác nhau thì thông tin của hình ảnh thứ nhất sẽ được lưu vào bảng posts và thông tin của hình ảnh từ thứ hai trở đi sẽ được lưu vào bảng posts\_items.

*A screenshot of a cell phone

Description automatically generated with medium confidence*

Hình 3.1.3‑6:Sơ đồ thuật toán cập nhập bài đăng mới theo ngày

Bảng 3.3‑2: Thực hiện việc cập nhập bài đăng mới theo ngày

|  |
| --- |
| import mysql.connector  from facebook\_scraper import get\_posts  from urllib.parse import urlparse, parse\_qs  import requests  from datetime import datetime, timedelta  # Kết nối tới cơ sở dữ liệu MySQL  mydb = mysql.connector.connect(      host="localhost",      user="root",      password="",      database="data\_clb"  )  mycursor = mydb.cursor()  # Số ngày cần cập nhật (ví dụ: cập nhật 7 ngày trước đây)  num\_days = 7  # Tính ngày bắt đầu cần cập nhật  target\_date = datetime.now().date() - timedelta(days=num\_days)  # Lấy danh sách các fanpage từ bảng fanpages  mycursor.execute("SELECT \* FROM fanpages")  fanpages = mycursor.fetchall()  # Lấy dữ liệu bài đăng trên Facebook và đồng bộ vào cơ sở dữ liệu MySQL  for fanpage in fanpages:      fanpage\_name = fanpage[0]      fanpage\_url = fanpage[1]  # Lấy dữ liệu bài đăng trên Facebook và đồng bộ vào cơ sở dữ liệu MySQL  for fanpage in fanpages:      for post in get\_posts(group=fanpage[1], pages=2):          # Kiểm tra ngày tạo bài đăng          post\_date = post['time'].date()          if post\_date < target\_date:              continue  # Bỏ qua nếu bài đăng cũ hơn ngày cần cập nhật          # Kiểm tra xem post\_id đã tồn tại trong cơ sở dữ liệu chưa          mycursor.execute("SELECT \* FROM posts WHERE post\_id = %s", (post['post\_id'],))          result = mycursor.fetchone()          if result is None:              # Thêm dữ liệu vào bảng `posts`              sql = "INSERT INTO posts (fanpage\_name, post\_id, user\_name, post\_content, created\_time, image) VALUES (%s, %s, %s, %s, %s, %s)"              val = (fanpage[1], post['post\_id'], post['username'], post['text'], post['time'], post['image'])              mycursor.execute(sql, val)              mydb.commit()              # Trích xuất ID hình ảnh và cập nhật vào cột fbid\_image (nếu có)              if post['image'] is not None:                  image\_id = None                  parsed\_url = urlparse(post['image'])                  parsed\_qs = parse\_qs(parsed\_url.query)                  if 'fbid' in parsed\_qs:                      image\_id = parsed\_qs['fbid'][0]                  sql\_update = "UPDATE posts SET fbid\_image = %s WHERE post\_id = %s"                  val\_update = (image\_id, post['post\_id'])                  mycursor.execute(sql\_update, val\_update)                  mydb.commit()              # Thêm dữ liệu vào bảng `posts\_items` cho các hình ảnh khác (nếu có)              if post['images'] is not None and len(post['images']) > 1:                  for i in range(1, len(post['images'])):                      sql = "INSERT INTO posts\_items (fanpage\_name, post\_id, image\_items, image\_content) VALUES (%s, %s, %s, %s)"                      val = (fanpage[1], post['post\_id'], post['images'][i], post['text'])                      mycursor.execute(sql, val)                      mydb.commit()                      # Trích xuất ID hình ảnh và cập nhật vào cột fbid\_image\_items (nếu có)                      if post['images'][i] is not None:                          image\_id\_items = None                          parsed\_url\_items = urlparse(post['images'][i])                          parsed\_qs\_items = parse\_qs(parsed\_url\_items.query)                          if 'fbid' in parsed\_qs\_items:                              image\_id\_items = parsed\_qs\_items['fbid'][0]                          sql\_update\_items = "UPDATE posts\_items SET fbid\_image\_items = %s WHERE image\_items = %s"                          val\_update\_items = (image\_id\_items, post['images'][i])                          mycursor.execute(sql\_update\_items, val\_update\_items)                          mydb.commit()      # Lấy danh sách các bài đăng và cập nhật nội dung hình ảnh vào cơ sở dữ liệu      mycursor.execute("SELECT post\_id, fbid\_image\_items FROM posts\_items")      posts\_items = mycursor.fetchall()      for post\_id, fbid\_image\_items in posts\_items:          if fbid\_image\_items and post\_id:              url = "https://m.facebook.com/story.php?story\_fbid={fbid}&id={user\_id}".format(                  fbid=fbid\_image\_items,                  user\_id=post\_id              )              response = requests.get(url)              if response.status\_code == 200:                  posts = list(get\_posts(post\_urls=[url], cookies="cookies.txt"))                  if len(posts) > 0:                      image\_content = posts[0]["post\_text"]                      # Cập nhật dữ liệu vào cơ sở dữ liệu                      sql\_update\_items = "UPDATE posts\_items SET image\_content = %s WHERE post\_id = %s"                      val\_update\_items = (image\_content, post\_id)                      mycursor.execute(sql\_update\_items, val\_update\_items)                      mydb.commit() |

**Bảng 3.3.2 :** Kết quả của đoạn code này là nếu trong khoảng thời gian 7 ngày mà có bài đăng mới thì sẽ cập nhập các bài đăng mới đó vào database. Chúng ta cũng có thể thay đổi thời gian cập nhập bài viết bằng cách thay đổi số ngày trong đó code trên.

VD: Nếu ta muốn câp nhập bài viết mới trong vòng 4 ngày thì ta thay đổi biến :

num\_days = 7 -> num\_days = 4

A screenshot of a computer

Description automatically generated with medium confidence

Hình 3.1.3‑7: Khi chưa chạy chương trình cập nhập theo ngày.

Hình 3.3-7: Cho ta thấy bài đăng mới nhất được lưu vào database là ngày 17-06-2023 vào lúc 22:44:32. Sau khi ta chạy đoạn code cập nhập các bài đăng mới nhất trong vòng 14 thì kết quả sẽ cho như hình dưới:

A screenshot of a computer

Description automatically generated with medium confidence

Hình 3.1.3‑8: Kết quả sau khi chạy code cập nhập theo ngày.

Kết quả sau khi chạy chương trình thì ta thấy trên hình là bài viết mới nhất được cập nhập là vào ngày 29-06-2023 vào lúc 21:37:03.

A screenshot of a chat

Description automatically generated with low confidence

Hình 3.1.3‑9: Sơ đồ thuật toán cập nhập lại nội dung bài đăng

Bảng 3.3‑3: Thực hiện việc cập nhập lại nội dung bài đăng

|  |
| --- |
| import mysql.connector  from facebook\_scraper import get\_posts  from urllib.parse import urlparse, parse\_qs  import requests  # Kết nối tới cơ sở dữ liệu MySQL  mydb = mysql.connector.connect(      host="localhost",      user="root",      password="",      database="data\_clb"  )  mycursor = mydb.cursor()  # Lấy danh sách các fanpage từ bảng fanpages  mycursor.execute("SELECT \* FROM fanpages")  fanpages = mycursor.fetchall()  # Lấy dữ liệu bài đăng trên Facebook và đồng bộ vào cơ sở dữ liệu MySQL  for fanpage in fanpages:      fanpage\_name = fanpage[0]      fanpage\_url = fanpage[1]      # Lấy dữ liệu bài đăng trên Facebook và đồng bộ vào cơ sở dữ liệu MySQL      for fanpage in fanpages:          for post in get\_posts(group=fanpage[1], pages=2):              # Kiểm tra xem post\_id đã tồn tại trong cơ sở dữ liệu chưa              mycursor.execute("SELECT \* FROM posts WHERE post\_id = %s", (post['post\_id'],))              result = mycursor.fetchone()              if result is None:                  # Thêm dữ liệu vào bảng `posts`                  sql = "INSERT INTO posts (fanpage\_name, post\_id, user\_name, post\_content, created\_time, image) VALUES (%s, %s, %s, %s, %s, %s)"                  val = (fanpage[1], post['post\_id'], post['username'], post['text'], post['time'], post['image'])                  mycursor.execute(sql, val)                  mydb.commit()              else:                  # Kiểm tra các trường thay đổi như nội dung và hình ảnh                  if result[4] != post['text'] or result[6] != post['image']:                      # Cập nhật dữ liệu mới vào bảng `posts`                      sql\_update = "UPDATE posts SET post\_content = %s, image = %s WHERE post\_id = %s"                      val\_update = (post['text'], post['image'], post['post\_id'])                      mycursor.execute(sql\_update, val\_update)                      mydb.commit()              # Trích xuất ID hình ảnh và cập nhật vào cột fbid\_image (nếu có)              if post['image'] is not None:                  image\_id = None                  parsed\_url = urlparse(post['image'])                  parsed\_qs = parse\_qs(parsed\_url.query)                  if 'fbid' in parsed\_qs:                      image\_id = parsed\_qs['fbid'][0]                  sql\_update = "UPDATE posts SET fbid\_image = %s WHERE post\_id = %s"                  val\_update = (image\_id, post['post\_id'])                  mycursor.execute(sql\_update, val\_update)                  mydb.commit()              # Thêm dữ liệu vào bảng `posts\_items` cho các hình ảnh khác (nếu có)              if post['images'] is not None and len(post['images']) > 1:                  for i in range(1, len(post['images'])):                      sql = "INSERT INTO posts\_items (fanpage\_name, post\_id, image\_items, image\_content) VALUES (%s, %s, %s, %s)"                      val = (fanpage[1], post['post\_id'], post['images'][i], post['text'])                      mycursor.execute(sql, val)                      mydb.commit()                      # Trích xuất ID hình ảnh và cập nhật vào cột fbid\_image\_items (nếu có)                      if post['images'][i] is not None:                          image\_id\_items = None                          parsed\_url\_items = urlparse(post['images'][i])                          parsed\_qs\_items = parse\_qs(parsed\_url\_items.query)                          if 'fbid' in parsed\_qs\_items:                              image\_id\_items = parsed\_qs\_items['fbid'][0]                          sql\_update\_items = "UPDATE posts\_items SET fbid\_image\_items = %s WHERE image\_items = %s"                          val\_update\_items = (image\_id\_items, post['images'][i])                          mycursor.execute(sql\_update\_items, val\_update\_items)                          mydb.commit()          # Lấy danh sách các bài đăng và cập nhật nội dung hình ảnh vào cơ sở dữ liệu          mycursor.execute("SELECT post\_id, fbid\_image\_items FROM posts\_items")          posts\_items = mycursor.fetchall()          for post\_id, fbid\_image\_items in posts\_items:              if fbid\_image\_items and post\_id:                  url = "https://m.facebook.com/story.php?story\_fbid={fbid}&id={user\_id}".format(                      fbid=fbid\_image\_items,                      user\_id=post\_id                  )                  response = requests.get(url)                  if response.status\_code == 200:                      posts = list(get\_posts(post\_urls=[url], cookies="cookies.txt"))                      if len(posts) > 0:                          image\_content = posts[0]["post\_text"]                          # Kiểm tra và cập nhật nội dung hình ảnh mới                          mycursor.execute("SELECT image\_content FROM posts\_items WHERE post\_id = %s", (post\_id,))                          result\_items = mycursor.fetchone()                          if result\_items[0] != image\_content:                              # Cập nhật dữ liệu vào cơ sở dữ liệu                              sql\_update\_items = "UPDATE posts\_items SET image\_content = %s WHERE post\_id = %s"                              val\_update\_items = (image\_content, post\_id)                              mycursor.execute(sql\_update\_items, val\_update\_items)                              mydb.commit() |

**Bảng 3.3.3:** Sau khi chạy đoạn code ta sẽ có kết quả là : Nếu trong 1 bài đăng mà ta có sự chỉnh sửa như : nội dung bài đăng, hình ảnh… thì ta sẽ cập nhập lại nội dung đã chỉnh sửa đó vào lại database mà không cần tạo dòng dữ liệu mới trong database.

A screenshot of a video game

Description automatically generated

Hình 3.1.3‑10: Nội dung ban đâu hình ảnh khi chưa chỉnh sửa trên facebook.A screenshot of a computer

Description automatically generated with medium confidence

Hình 3.1.3‑11: Nội dung ban đâu hình ảnh khi chưa chỉnh sửa trong database

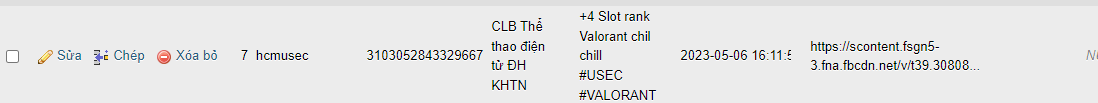
A screenshot of a video game

Description automatically generated

Hình 3.1.3‑12: Sau chi chỉnh sửa nội dung trên facebook.

**Kết quả sau khi chạy chương trình:**

Nội dung của bài đăng đó đã được cập nhập lại trong database.

****

Hình 3.1.3‑13: Kết quả sau khi chạy chương trình

A screenshot of a phone

Description automatically generated with medium confidence

Hình 3.1.3‑14: Sơ đồ thuật toán cập nhập lại URL hình ảnh bị lỗi

Bảng 3.3‑4: Thực hiện việc cập nhập lại URL hình ảnh khi bị lỗi

|  |
| --- |
| import mysql.connector  from facebook\_scraper import get\_posts  # Kết nối đến cơ sở dữ liệu  mydb = mysql.connector.connect(      host="localhost",      user="root",      password="",      database="data\_clb"  )  mycursor = mydb.cursor()  # Lấy danh sách post\_id từ cơ sở dữ liệu  mycursor.execute("SELECT post\_id, fbid\_image FROM posts")  posts = mycursor.fetchall()  mycursor.execute("SELECT post\_id, fbid\_image\_items FROM posts\_items")  post\_items = mycursor.fetchall()  # Đọc file cookies  with open("C:/Users\quant/OneDrive/Máy tính/Workspace/Seminar tốt nghiệp/Đồng bộ dữ liệu/cookies.txt", "r") as f:      cookies = f.read()  # Duyệt qua từng post\_id và lấy url của hình ảnh bằng facebook\_scraper  for post\_id, fbid\_image in posts:      if fbid\_image and post\_id:          url = "https://m.facebook.com/story.php?story\_fbid={fbid}&id={user\_id}".format(              fbid=fbid\_image,              user\_id=post\_id          )          posts = list(get\_posts(post\_urls=[url], cookies="cookies.txt"))          if len(posts) > 0:              image\_url = posts[0]["image"]              # Cập nhật dữ liệu vào cơ sở dữ liệu              sql = "UPDATE posts SET image = %s WHERE post\_id = %s"              val = (image\_url, post\_id)              mycursor.execute(sql, val)              mydb.commit()  for post\_id, fbid\_image\_items in post\_items:      if fbid\_image\_items and post\_id:          url = "https://m.facebook.com/story.php?story\_fbid={fbid}&id={user\_id}".format(              fbid=fbid\_image\_items,              user\_id=post\_id          )          posts = list(get\_posts(post\_urls=[url], cookies="cookies.txt"))          if len(posts) > 0:              image\_url\_items = posts[0]["image"]              # Cập nhật dữ liệu vào cơ sở dữ liệu              sql = "UPDATE posts\_items SET image\_items = %s WHERE post\_id = %s"              val = (image\_url\_items, post\_id)              mycursor.execute(sql, val)              mydb.commit() |

**Bảng 3.3.4:** Sau khi ta chạy đoạn code trên thì sẽ cho ta kết quả là: Khi ta cập nhập dữ liệu về 2 bảng posts và posts\_items thì các bài viết có định dạng hình ảnh là m.facebook thì bị lỗi hiển thị lên trang web như hình 3.3.4.1 dưới:

Ảnh có chứa văn bản, ảnh chụp màn hình, Trang web, phần mềm

Mô tả được tạo tự động

Hình 3.1.3‑15: Khi URL hình ảnh bị lỗi.

Sau khi chạy đoạn code trên thì sẽ cập nhập lại URL hình ảnh để hiển thị lại trên web không bị lỗi như hình 3.3.4.2 bên dưới:

Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Trang web

Mô tả được tạo tự động

Hình 3.1.3‑16: Kết quả sau khi sửa lỗi hình ảnh.

## Dùng mô hình MVC trong Laravel để lấy dữ liệu từ MySQL về giao diện.

### Taọ Model

* **PostModel.php**

Bảng 3.4‑1: postModel

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class PostModel extends Model  {      //gọi table 'posts'      use HasFactory;      protected $table ='posts';      public function getImages\_Post()      {          return $this -> hasMany('App\Models\images\_Post\_Model','post\_id','post\_id');      }      public function getImages\_Null()      {          return $this -> hasOne('App\Models\CLB\_Model','name','fanpage\_name');      }  } |

* **images\_Post\_Model.php**

Bảng 3.4‑2: images\_Post\_Model

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class images\_Post\_Model extends Model  {      use HasFactory;      protected $table='posts\_items';  } |

* **ThanhVienModel.php**

Bảng 3.4‑3: ThanhVienModel

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class ThanhVienModel extends Model  {      //gọi table 'thanhvien'      use HasFactory;      protected $table ='thanhvien';  } |

* caulacboModel.php

Bảng 3.4‑4: caulacboModel

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class caulacboModel extends Model  {      use HasFactory;      // gọi table 'clb' để truyền one to many      protected $table = 'fanpages';      public function getThanhVien()      {          return $this -> hasMany('App\Models\ThanhVienModel','clb\_id');      }  } |

* CLB\_Model.php

Bảng 3.4‑5: CLB\_Model

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class CLB\_Model extends Model  {      //gọi table 'fanpages'      use HasFactory;      protected $table = 'fanpages';  } |

### Tạo Controller

Bảng 3.4‑6: IndexController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use App\Models\PostModel;  use App\Models\caulacboModel;  use App\Models\baiviet\_Model;  use App\Models\ThanhVienModel;  use App\Models\CLB\_Model;  use App\Models\images\_Post\_Model;  use Illuminate\Http\RedirectResponse;  use Illuminate\Http\Response;  use Illuminate\Support\Carbon;  class IndexController extends Controller  {      // gọi model PostModel, CLB\_Model về index (gồm có slide, posts, tên clb)      public function show()      {      }      //search      public function search(Request $request)      {          // slide, tên clb          $data1 = CLB\_Model::all();          //search          $search = $request->search;          $dateFilter = $request->date\_filter;  //filter ngày tháng cho bài posts          $selectedClub = $request->club\_filter;   //lọc câu lạc bộ          $query = PostModel::query()              ->Where(function ($query) use ($search) {                  $query->where('post\_content', 'like', "%$search%");              })              ->Where(function ($query) use ($dateFilter) {                  switch ($dateFilter) {                      case 'day':                          $query->whereDate('created\_time', Carbon::today());                          break;                      case 'week':                          $query->whereBetween('created\_time', [Carbon::now()->startOfWeek(), Carbon::now()->endOfWeek()]);                          break;                      case 'month':                          $query->whereMonth('created\_time', Carbon::now()->month);                          break;                      case 'year':                          $query->whereYear('created\_time', Carbon::now()->year);                          break;                  }              })              ->Where(function ($query) use ($selectedClub) {                  switch ($selectedClub) {                      case 'hcmusec':                          $query->where('fanpage\_name', 'hcmusec');                          break;                      case 'escussh':                          $query->where('fanpage\_name', 'escussh');                          break;                  }              });          $data = $query->get()->sortByDesc('created\_time');          return view('frontend.index', compact('data', 'search', 'data1', 'dateFilter', 'selectedClub'));      }      // chi tiết bài posts      function show1($postId)      {          $data1 = CLB\_Model::all();          $post = PostModel::find($postId);          $post\_2 =  PostModel::find($postId)->getImages\_Null;          $post\_1 = PostModel::find($postId)->getImages\_Post;          return view('frontend.baiviet', compact('post', 'post\_1', 'post\_2','data1' ));      }      //chi tiết trang giới thiệu      public function gioithieu($gioithieuId)      {          $data1 = CLB\_Model::all();          $gioithieu = caulacboModel::find($gioithieuId)->getThanhVien;          $gioithieu1 = CLB\_Model::find($gioithieuId);          return view('frontend.gioithieu', compact('gioithieu', 'gioithieu1', 'data1'));      }  } |

### Tạo Views

Bảng 3.4‑7: web.php

|  |
| --- |
| <?php  use Illuminate\Support\Facades\Route;  use App\Http\Controllers\IndexController;  use App\Http\Controllers\BaiVietController;  // truyền /index về giao diện  Route::get ('/', [IndexController::class, 'show']) -> name('home');  // truyền chi tiết bài posts  Route::get('post/{post}',[IndexController::class, 'show1']);  // truyền chi tiết giới thiệu clb  Route::get('gioithieu/{gioithieu}',[IndexController::class, 'gioithieu']);  //search  Route::get('/search',[IndexController::class,'search'])-> name('HOME1'); |

## Kết luận

Trong chương này, chúng em đã tiến hành một loạt các bước để xây dựng một hệ thống đáng tin cậy để lưu trữ và hiển thị dữ liệu từ Facebook trên trang web của chúng em. Sử dụng thư viện python facebook\_scraper, chúng em đã phát triển một công cụ mạnh mẽ cho việc đồng bộ và lưu trữ dữ liệu từ các bài đăng trên Facebook. Chúng em đã tạo ra các chức năng để tự động cập nhật những bài đăng mới nhất theo khoảng thời gian cố định, đảm bảo rằng người dùng luôn được cung cấp thông tin mới nhất.

Ngoài ra, chúng em đã xử lý các vấn đề như cập nhật lại nội dung bài đăng nếu có sự chỉnh sửa và sửa lỗi đường dẫn hình ảnh bị lỗi. Điều này đảm bảo rằng dữ liệu hiển thị trên trang web của chúng em luôn chính xác và hấp dẫn đối với người dùng. Với việc sử dụng Laravel, một framework phát triển web mạnh mẽ, chúng em đã dễ dàng và linh hoạt trong việc hiển thị thông tin từ cơ sở dữ liệu trên trang web.

Bằng cách sử dụng công nghệ và công cụ mạnh mẽ, chúng em đã tạo ra một hệ thống đáng tin cậy và hiệu quả để lưu trữ và hiển thị dữ liệu từ Facebook trên trang web của chúng em. Điều này giúp chúng em mang đến trải nghiệm tốt nhất cho người dùng, đảm bảo rằng họ luôn được cung cấp thông tin mới nhất và chính xác nhất từ các bài đăng trên Facebook.

# KẾT QUẢ THỰC NGHIỆM

Trong chương này, chúng em trình bày cấu trúc dữ liệu của hệ thống, bao gồm các bảng trong cơ sở dữ liệu và các tệp tin liên quan đến website. Chúng em cũng trình bày kết quả thu được sau quá trình triển khai website, bao gồm trang chủ, trang chi tiết bài viết và trang giới thiệu câu lạc bộ. Mục tiêu của chúng em là cung cấp một cái nhìn tổng quan về cấu trúc và hiệu quả của website trong phạm vi nghiên cứu.

## Tổ chức

### Database

Gồm các bảng:

* fanpages: Chứa fanpage của câu lạc bộ. 1 fanpage chứa các field : id, name, url, HinhAnh, Gioithieu, Ten.
* Posts: Chứa thông tin bài đăng. 1 bài đăng chứa các field: id, fanpage\_name, post\_id, user\_name, post\_content, created\_time, image, fbid\_image.
* Posts\_items: Chứa các thông tin hình ảnh tiếp theo nếu bài đăng có nhiều hình ảnh. Mỗi hình ảnh đó chứa các field : id, fanpage\_name, post\_id, image\_items, image\_content, fbid\_image\_items.
* thanhvien: Chứa các thành viên trong câu lạc bộ. Mỗi thành viên chứa các field: id, HinhThe, HovaTen, Khoa, NamHoc, ChucVu, fanpage\_id.

### Website

File **index.blade.php:** hiện danh sách câu lạc bộ, bài posts các câu lạc bộ, hình ảnh, lọc câu lạc bộ, lọc ngày tháng, tìm kiếm từ khóa.

File **baiviet.blade.php:** Hiển thị chi tiết bài viết.

File **gioithieu.blade.php:** Hiển thị hình ảnh câu lạc bộ, giới thiệu câu lạc bộ và thành viên.

A picture containing screenshot, line, font, diagram

Description automatically generated

Hình 4.1.2‑1: Sơ đồ hoạt động của user khi truy cập website.

Hình 4.1.2-1: Sơ đồ trình bày các thành phần chính và luồng thông tin giữa chúng trong kiến trúc của trang web quản lý thông tin của các câu lạc bộ trên Facebook.

## Kết quả

### Trang chủ:

* **Header và slide:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 4.2.1‑1: Header và slider trang chủ.

* **Tìm kiếm và bộ lọc thời gian:**

*Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động*

Hình 4.2.1‑2: Tìm kiếm và bộ lọc thời gian.

* **Tin tức cập nhật và sidebar:**

*Ảnh có chứa văn bản, Website, Quảng cáo trực tuyến, Trang web

Mô tả được tạo tự động*

Hình 4.2.1‑3: Tin tức mới cập nhập.

*Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Trang web

Mô tả được tạo tự động*

Hình 4.2.1‑4: sidebar

* **footer:**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình 4.2.1‑5: Footer chung của tất cả các trang

### Trang chi tiết bài viết:

**Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Trang web

Mô tả được tạo tự động**

Hình 4.2.2‑1: Trang chi tiết bài viết.

### Trang giới thiệu câu lạc bộ

**Ảnh có chứa văn bản, Phông chữ, Trang web, phần mềm

Mô tả được tạo tự động**

Hình 4.2.3‑1: Phần giới thiệu CLB của trang giới thiệu.

*Ảnh có chứa văn bản, ảnh chụp màn hình, trái cây

Mô tả được tạo tự động*

Hình 4.2.3‑2: Thành viên của CLB.A screenshot of a phone

Description automatically generated with medium confidenceA screenshot of a cellphone

Description automatically generated with medium confidence

## Kết luận

Chúng em đã mô tả cấu trúc database bao gồm các bảng "fanpages", "Posts", "Posts\_items" và "thanhvien", lưu trữ thông tin về fanpage, bài đăng, hình ảnh và thành viên. Trang web được triển khai với trang chủ hiển thị danh sách câu lạc bộ, bài đăng, hình ảnh và công cụ tìm kiếm. Trang chi tiết bài viết và trang giới thiệu câu lạc bộ cung cấp thông tin chi tiết về bài viết, câu lạc bộ và thành viên.

Kết quả thực nghiệm cho thấy dự án đã hoàn thành thành công và đáp ứng yêu cầu cơ bản ban đầu. Giao diện người dùng thân thiện và dễ sử dụng, với các chức năng tìm kiếm, lọc và xem chi tiết bài viết hoạt động hiệu quả. Thông tin về câu lạc bộ, bài đăng và thành viên được hiển thị rõ ràng và hấp dẫn.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

*Chương này tóm tắt nội dung đề tài và đề xuất hướng phát triển tương lai. Website đã hoàn thành các yêu cầu cơ bản và có giao diện thân thiện. Hướng phát triển tiềm năng bao gồm tăng cường tính bảo mật, mở rộng đồng bộ dữ liệu từ fanpage và cải thiện giao diện để tạo trải nghiệm người dùng tốt hơn. Mục tiêu là nâng cao chất lượng và hấp dẫn của trang thông tin.*

## Kết luận

Ba tháng trôi qua, một khoảng thời gian không dài, nhưng với sự hướng dẫn tận tâm của thầy Trương Tấn Quang và anh Nguyễn Mạnh Tiến, đề tài xây dựng trang web quản lý thông tin các câu lạc bộ trên Facebook của chúng em đã được hoàn thành.

Trong quá trình thực hiện đề tài, chúng em đã gặp những thuận lợi và khó khăn, nhưng cuối cùng chúng em đã hoàn thiện một phiên bản cơ bản của trang web, mặc dù không tránh được những sai sót nhỏ. Tuy nhiên, chúng em đã đạt được những mục tiêu chính đã đề ra:

* Phân tích và đánh giá yêu cầu cơ bản của trang web: Chúng em đã nắm bắt và hiểu được những yêu cầu quan trọng để xây dựng trang web quản lý thông tin các câu lạc bộ trên Facebook.
* Xây dựng giao diện trang web sử dụng HTML và CSS: Chúng em đã áp dụng kiến thức về HTML và CSS để tạo ra giao diện trực quan và thân thiện với người dùng.
* Sử dụng Python để lấy dữ liệu từ Facebook và lưu vào cơ sở dữ liệu: Chúng em đã sử dụng thư viện facebook\_scraper của Python để truy xuất thông tin từ Facebook và lưu trữ vào cơ sở dữ liệu để dễ dàng quản lý.
* Áp dụng mô hình MVC của Laravel: Chúng em đã sử dụng mô hình MVC (Model-View-Controller) của Laravel để trích xuất dữ liệu từ cơ sở dữ liệu và hiển thị nó trên trang web.
* Sử dụng hiệu quả các công cụ như Xampp và Visual Studio Code: Chúng em đã sử dụng Xampp và Visual Studio Code để hỗ trợ quá trình lập trình và triển khai trang web.

Kết quả cuối cùng là chúng em đã hoàn thành phần giao diện trang chủ, tin tức chi tiết và phần giới thiệu. Các phần này được thiết kế với giao diện thân thiện, giúp người dùng dễ dàng tiếp cận thông tin về các câu lạc bộ một cách nhanh chóng và chính xác hơn. Điều này giúp người dùng dễ dàng tìm kiếm thông tin và nắm bắt những nội dung quan trọng.

Tuy chúng em nhận thấy còn một số sai sót trong quá trình thực hiện, nhưng chúng em rất tự hào về thành tựu đã đạt được. Đề tài này đã mang lại cho chúng em nhiều kiến thức và kỹ năng mới trong việc phát triển trang web. Chúng em sẽ tiếp tục hoàn thiện và nâng cao chất lượng trang web này để mang lại trải nghiệm tốt nhất cho người dùng.

## Hướng phát triển

Ngày nay, Internet đã trở thành một nguồn thông tin khổng lồ và cung cấp sự tiện ích cho mọi người trên toàn cầu. Việc học và tìm kiếm thông tin trực tuyến đã trở thành một xu hướng phổ biến. Chúng em hiểu rõ tầm quan trọng của việc cung cấp thông tin chất lượng và luôn cố gắng nâng cao chất lượng website của chúng em. Chúng em đặt ra một số mục tiêu cụ thể để đạt được điều này.

Mục tiêu đầu tiên của chúng em là tăng cường tính bảo mật và quản lý thông tin bằng cách thêm hệ thống phân quyền cho người dùng trên website. Bằng cách xác định và phân loại các loại người dùng khác nhau, chúng em sẽ có thể xác định quyền hạn và vai trò tương ứng cho từng người dùng. Điều này sẽ giúp tạo ra một môi trường an toàn và chuyên nghiệp, đồng thời đảm bảo rằng người dùng chỉ có quyền truy cập và sử dụng các thông tin phù hợp với vai trò của họ.

Thứ hai, chúng em muốn mở rộng khả năng đồng bộ dữ liệu trên website của mình. Hiện tại, chúng em chỉ đồng bộ dữ liệu từ các nhóm (group), nhưng chúng em muốn mở rộng để có thể đồng bộ dữ liệu từ cả các fanpage. Để làm điều này, chúng em sẽ sử dụng các API của các nền tảng mạng xã hội để lấy thông tin từ fanpage và hiển thị trực tiếp trên website của chúng em. Điều này sẽ mang đến sự cập nhật và phong phú hơn cho người dùng, cho phép họ truy cập vào nhiều nguồn thông tin và nội dung đa dạng.

Cuối cùng, chúng em muốn cải thiện giao diện của website để làm nó trở nên hấp dẫn hơn. Chúng em sẽ nghiên cứu các xu hướng thiết kế giao diện hiện đại và chuyên nghiệp, nhằm tạo ra một giao diện đơn giản, trực quan và dễ sử dụng. Chúng em sẽ tận dụng màu sắc, hình ảnh và biểu đồ một cách tối ưu, nhằm tạo nên một trải nghiệm thú vị và trực quan cho người dùng. Bằng cách thúc đẩy sự tương tác và tạo điểm nhấn hợp lý, chúng em hy vọng sẽ thu hút người dùng và tạo ra ấn tượng sâu sắc hơn với website của chúng em.

Với những mục tiêu này, chúng em tin rằng website của chúng em sẽ ngày càng hoàn thiện hơn, mang đến cho người dùng một trang thông tin chất lượng, chuyên nghiệp và hấp dẫn.

# TÀI LIỆU THAM KHẢO

1. CSS Tutorial. (n.d.). W3Schools. Retrieved June 18, 2023, from https://www.w3schools.com/css/default.asp
2. Extract direct link for image and posts · Issue #213 · kevinzg/facebook-scraper. (2021, April 21). GitHub. Retrieved June 18, 2023, from https://github.com/kevinzg/facebook-scraper/issues/213
3. HTML Tutorial. (n.d.). W3Schools. Retrieved June 18, 2023, from <https://www.w3schools.com/html/default.asp>
4. Download · Bootstrap. (n.d.). Bootstrap. Retrieved June 16, 2023, from <https://getbootstrap.com/docs/4.3/getting-started/download/>
5. BOOTSTRAP.MIN.JS: DOWNLOAD - CDNPKG. (n.d.). CDNPKG .com. Retrieved June 16, 2023, from <https://www.cdnpkg.com/bootswatch/file/bootstrap.min.js/>
6. JQUERY.SLICKNAV.MIN.JS: DOWNLOAD - CDNPKG. (n.d.). CDNPKG .com. Retrieved June 16, 2023, from <https://www.cdnpkg.com/SlickNav/file/jquery.slicknav.min.js/>
7. kevinzg/facebook-scraper: Scrape Facebook public pages without an API key. (n.d.). GitHub. Retrieved June 18, 2023, from https://github.com/kevinzg/facebook-scraper
8. The PHP Framework For Web Artisans. (n.d.). Laravel. Retrieved June 18, 2023, from https://laravel.com/docs/10.x
9. PHP Tutorial. (n.d.). W3Schools. Retrieved June 18, 2023, from <https://www.w3schools.com/php/default.asp>