

TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ NGÀNH**  
**HỌC KỲ I, NĂM HỌC 2025-2026**

**THIẾT KẾ HỆ THỐNG GHI CHÚ CÁ NHÂN TRỰC  
TUYẾN ỨNG DỤNG CÔNG NGHỆ WEB ĐỘNG**

*Giảng viên hướng dẫn:*

ThS. Nguyễn Ngọc Đan Thanh

*Sinh viên thực hiện:*

Họ tên: Trương Truyền Phúc Minh

MSSV: 110123263

Lớp: DA23TTD

**Vĩnh Long, tháng 01 năm 2026**

TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ NGÀNH**  
**HỌC KỲ I, NĂM HỌC 2025-2026**

**THIẾT KẾ HỆ THỐNG GHI CHÚ CÁ NHÂN TRỰC  
TUYẾN ỨNG DỤNG CÔNG NGHỆ WEB ĐỘNG**

*Giảng viên hướng dẫn:*

ThS. Nguyễn Ngọc Đan Thanh

*Sinh viên thực hiện:*

Họ tên: Trương Truyền Phúc Minh

MSSV: 110123263

Lớp: DA23TTD

**Vĩnh Long, tháng 01 năm 2026**

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

(Ký tên và ghi rõ họ tên)

**TRANG NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Vĩnh Long, ngày ... tháng ... năm 2026*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

## LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn sâu sắc đến quý thầy cô khoa Công nghệ Thông tin, đặc biệt là giảng viên hướng dẫn em – cô Nguyễn Ngọc Đan Thanh, người đã tận tình hướng dẫn, hỗ trợ và khơi dậy tư duy hệ thống cho em trong suốt quá trình thực hiện đồ án vừa qua.

Nhờ sự hướng dẫn nhiệt tình của cô, em đã có cơ hội tiếp cận và hiểu sâu hơn về công nghệ web – một lĩnh vực không chỉ mang tính học thuật mà còn có ứng dụng vô cùng thiết thực trong thực tế.

Mặc dù đồ án có thể chưa hoàn hảo, nhưng em mong đây sẽ là dấu mốc giúp em rèn luyện tư duy phân tích và sẵn sàng cho những đồ án tiếp theo.

Một lần nữa, em xin chân thành cảm ơn cô và kính chúc cô luôn mạnh khỏe, công tác tốt, và tiếp tục truyền cảm hứng cho các thế hệ sinh viên tiếp theo.

## MỤC LỤC

Trang nhận xét của giảng viên hướng dẫn.....	i
Trang nhận xét của thành viên hội đồng .....	ii
LỜI CẢM ƠN.....	iii
MỤC LỤC.....	iv
DANH MỤC HÌNH ẢNH.....	ix
DANH MỤC BẢNG BIỂU .....	xi
Tóm tắt đồ án.....	xii
MỞ ĐẦU .....	1
Lý do chọn đề tài.....	1
Mục đích nghiên cứu .....	1
Đối tượng nghiên cứu.....	2
Phương pháp nghiên cứu .....	2
Phạm vi nghiên cứu .....	3
CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU .....	4
1.1 Tổng quan về vấn đề nghiên cứu .....	4
1.2 Một số hệ thống ghi chú phổ biến .....	4
1.2.1 Tổng quan về Google Keep .....	4
1.2.2 Phân tích Evernote .....	5
1.2.3 Giới thiệu về Notion.....	5
1.3 Ưu điểm của hệ thống.....	5
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT .....	7
2.1 Tổng quan về công nghệ web .....	7
2.1.1 Kiến trúc Client-Server.....	7
2.1.2 Quy trình hoạt động.....	8

2.2 Ngôn ngữ và công nghệ sử dụng.....	9
2.2.1 HTML – Ngôn ngữ đánh dấu siêu văn bản .....	10
2.2.2 CSS – Ngôn ngữ định dạng .....	10
2.2.2.1 <i>Inline style sheet</i> .....	11
2.2.2.2 <i>Internal style sheet</i> .....	11
2.2.2.3 <i>External style sheet</i> .....	12
2.2.3 JavaScript – Ngôn ngữ lập trình web .....	12
2.2.4 Mối quan hệ giữa HTML – CSS – JavaScript.....	13
2.2.5 Tổng quan về Bootstrap.....	13
2.2.5.1 <i>Bootstrap Containers</i> .....	14
2.2.5.2 <i>Bootstrap Buttons</i> .....	14
2.2.5.3 <i>Bootstrap Alerts</i> .....	15
2.2.5.4 <i>Bootstrap Tables</i> .....	15
2.2.5.5 <i>Bootstrap Navbar</i> .....	16
2.2.5.6 <i>Bootstrap Forms</i> .....	17
2.2.5.7 <i>Bootstrap Grid</i> .....	17
2.2.6 MySQL – Hệ quản trị cơ sở dữ liệu .....	18
2.2.6.1 <i>MySQL Create Table</i> .....	23
2.2.6.2 <i>MySQL Insert Data</i> .....	23
2.2.6.3 <i>MySQL Select Data</i> .....	23
2.2.6.4 <i>MySQL Where</i> .....	23
2.2.6.5 <i>MySQL Order By</i> .....	24
2.2.6.6 <i>MySQL Delete Data</i> .....	24
2.2.6.7 <i>MySQL Update Data</i> .....	24
2.2.7 PHP – Công nghệ xử lý dữ liệu và phát triển web động.....	25
2.2.7.1 <i>Biến trong PHP</i> .....	25

2.2.7.2 Câu lệnh <i>if</i> .....	25
2.2.7.3 Câu lệnh vòng lặp <i>for</i> .....	27
2.2.7.4 Câu lệnh vòng lặp <i>while</i> .....	27
2.2.7.5 Câu lệnh vòng lặp <i>do...while</i> .....	27
2.2.7.6 Hàm trong <i>PHP</i> .....	27
2.2.7.7 Lập trình hướng đối tượng <i>PHP – PHP OOP</i> .....	28
2.2.8 Quản lý dữ liệu với <i>PHP</i> và <i>MySQL</i> .....	30
2.2.8.1 Kết nối <i>csdl</i> .....	30
2.2.8.2 Các thao tác trên <i>csdl</i> .....	31
2.3 Tổng kết chương.....	32
<b>CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU .....</b>	<b>33</b>
3.1 Mô tả bài toán.....	33
3.2 Đặc tả yêu cầu hệ thống.....	33
3.2.1 Yêu cầu chức năng .....	34
3.2.2 Yêu cầu phi chức năng .....	34
3.3 Thiết kế dữ liệu.....	35
3.4 Mô hình dữ liệu .....	35
3.4.1 Danh sách các thực thể và mối kết hợp.....	35
3.4.2 Chi tiết các thực thể và mối kết hợp.....	36
3.5 Thiết kế xử lý .....	38
3.5.1 Sơ đồ phân rã chức năng .....	38
3.5.2 Mô hình DFD .....	39
3.5.2.1 Mô hình DFD mức ngữ cảnh.....	39
3.5.2.2 Mô hình DFD mức 1.....	40
3.5.2.3 Mô hình DFD mức 2.....	41
3.5.3 Thiết kế xử lý hệ thống.....	41



3.5.3.1 Đăng ký và đăng nhập.....	41
3.5.3.2 Xử lý ghi chú .....	42
3.5.3.3 Xử lý nhãn.....	42
3.5.3.4 Tìm kiếm ghi chú.....	42
3.6 Thiết kế giao diện .....	42
3.6.1 Giao diện đăng ký và đăng nhập.....	43
3.6.2 Giao diện danh sách ghi chú .....	44
3.6.3 Giao diện tạo và chỉnh sửa ghi chú .....	45
3.6.4 Giao diện xem lịch sử.....	45
3.6.5 Giao diện quản lý nhãn.....	46
3.6.6 Giao diện gán nhãn.....	46
3.6.7 Giao diện thùng rác .....	47
3.7 Kết chương.....	47
<b>CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU .....</b>	<b>48</b>
4.1 Kết quả xây dựng giao diện hệ thống.....	48
4.1.1 Giao diện đăng ký và đăng nhập.....	48
4.1.2 Giao diện tạo ghi chú.....	50
4.1.3 Giao diện danh sách ghi chú .....	50
4.1.4 Giao diện quản lý nhãn ghi chú .....	51
4.1.5 Giao diện sửa ghi chú .....	51
4.1.6 Giao diện xem lịch sử ghi chú .....	52
4.1.7 Giao diện gán nhãn ghi chú .....	52
4.1.8 Giao diện thùng rác chứa ghi chú .....	53
4.2 Kết quả xây dựng các chức năng chính.....	53
4.2.1 Chức năng quản lý tài khoản người dùng.....	54
4.2.2 Chức năng quản lý ghi chú .....	54

<b>Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động</b>	
4.2.3 Chức năng lưu lịch sử.....	55
4.3 Đánh giá kết quả đạt được .....	55
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>56</b>
5.1 Kết luận.....	56
5.2 Hướng phát triển.....	56
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>58</b>
<b>PHỤ LỤC (Nếu có) .....</b>	<b>59</b>

**DANH MỤC HÌNH ẢNH**

Hình 1.1 Google Keep.....	4
Hình 1.2 Evernote .....	5
Hình 1.3 Notion.....	5
Hình 2.1 Mô hình Client–Server.....	7
Hình 2.2 Phân loại lệnh SQL .....	19
Hình 3.1 Mô hình ERD.....	38
Hình 3.2 Sơ đồ phân rã chức năng .....	39
Hình 3.3 Mô hình DFD mức ngữ cảnh.....	39
Hình 3.4 Mô hình DFD mức 1 .....	40
Hình 3.5 Mô hình DFD mức 2.....	41
Hình 3.6 Sitemap hệ thống ghi chú cá nhân .....	43
Hình 3.7 Phác thảo giao diện đăng ký và đăng nhập .....	43
Hình 3.8 Phác thảo giao diện danh sách ghi chú .....	44
Hình 3.9 Phác thảo giao diện tạo và chỉnh sửa.....	45
Hình 3.10 Phác thảo giao diện xem lịch sử .....	45
Hình 3.11 Phác thảo giao diện quản lý nhãn .....	46
Hình 3.12 Phác thảo giao diện gán nhãn .....	46
Hình 3.13 Phác thảo giao diện thùng rác.....	47
Hình 4.1 Giao diện đăng ký .....	48
Hình 4.2 Giao diện đăng nhập .....	49
Hình 4.3 Giao diện chính.....	49
Hình 4.4 Giao diện tạo ghi chú .....	50
Hình 4.5 Giao diện danh sách ghi chú.....	50
Hình 4.6 Giao diện quản lý nhãn ghi chú.....	51

Hình 4.7 Giao diện sửa ghi chú.....	51
Hình 4.8 Giao diện xem lịch sử .....	52
Hình 4.9 Giao diện gán nhãn .....	52
Hình 4.10 Giao diện thùng rác.....	53
Hình 4.11 Chức năng quản lý tài khoản.....	54
Hình 4.12 Chức năng quản lý ghi chú.....	54
Hình 4.13 Chức năng lưu lịch sử ghi chú.....	55

**DANH MỤC BẢNG BIỂU**

Bảng 2.1 Các lệnh SQL DDL .....	19
Bảng 2.2 Các lệnh SQL DQL .....	20
Bảng 2.3 Các lệnh SQL DML.....	21
Bảng 2.4 Các lệnh SQL DCL .....	21
Bảng 2.5 Các lệnh SQL TCL.....	22
Bảng 3.1 Các yêu cầu chức năng .....	34
Bảng 3.2 Các yêu cầu phi chức năng .....	34
Bảng 3.3 Danh sách các thực thể và mối kết hợp .....	35
Bảng 3.4 Bảng users .....	36
Bảng 3.5 Bảng notes.....	36
Bảng 3.6 Bảng tags.....	37
Bảng 3.7 Bảng note_tags .....	37
Bảng 3.8 Bảng note_history.....	38

## TÓM TẮT ĐỒ ÁN

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, nhu cầu về một hệ thống lưu trữ và quản lý ghi chú cá nhân trực tuyến, tiện lợi, an toàn, có khả năng truy cập đa nền tảng ngày càng cấp thiết. Con người ngày nay có nhu cầu ghi chú, theo dõi công việc, lưu trữ dữ liệu cá nhân một cách tiện lợi, an toàn và có thể truy cập mọi lúc, mọi nơi.

Đề tài “Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động” được thực hiện nhằm xây dựng một website hỗ trợ người dùng tạo, chỉnh sửa, xóa, tìm kiếm và quản lý ghi chú trực tuyến. Hệ thống được phát triển với mục tiêu mang lại một công cụ quản lý thông tin tiện ích, dễ sử dụng, hoạt động ổn định trong môi trường web mà không cần cài đặt phần mềm.

Hướng tiếp cận của đề tài tập trung vào việc áp dụng các công nghệ web động với HTML, CSS, JavaScript, Bootstrap kết hợp cùng với PHP và MySQL. Dữ liệu được thiết kế theo mô hình quan hệ với 5 bảng chính gồm: users, notes, tags, note\_tags, note\_history, đảm bảo tính toàn vẹn và dễ mở rộng. Hệ thống cho phép người dùng đăng ký, đăng nhập, quản lý ghi chú cá nhân, gắn thẻ phân loại, lưu lại lịch sử chỉnh sửa, và hỗ trợ xóa tạm thời để phục hồi dữ liệu khi cần.

Kết quả đạt được của đề tài là một website ghi chú trực tuyến có giao diện thân thiện, dễ sử dụng, hoạt động ổn định trong môi trường XAMPP, có thể triển khai dễ dàng trên các máy chủ web khác. Ứng dụng đáp ứng tốt các yêu cầu cơ bản về quản lý ghi chú cá nhân, đồng thời có khả năng mở rộng để tích hợp các tính năng nâng cao như đồng bộ dữ liệu đám mây, chia sẻ ghi chú, và đăng nhập bằng tài khoản mạng xã hội trong tương lai.

## MỞ ĐẦU

### Lý do chọn đề tài

Trong thời đại công nghệ số hiện nay, các ứng dụng ghi chú cá nhân trên điện thoại như Samsung Notes đã trở nên phổ biến. Tuy nhiên, đa số các ứng dụng này hoạt động chủ yếu ở chế độ offline, việc đồng bộ dữ liệu giữa các thiết bị hoặc truy cập từ trình duyệt web còn hạn chế. Ví dụ như ta ghi chú ở nhà bằng laptop nhưng lỡ khi ra đường muốn sửa ghi chú bằng điện thoại lại không được.

Từ thực tế đó, em lựa chọn đề tài “Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động” nhằm:

Tạo ra một hệ thống ghi chú có thể truy cập mọi lúc, mọi nơi thông qua trình duyệt web.

Tìm hiểu và ứng dụng các công nghệ web front-end (HTML, CSS, JavaScript) và back-end (PHP, MySQL) trong việc xây dựng ứng dụng thực tế.

Nâng cao kỹ năng lập trình web, kỹ năng quản lý cơ sở dữ liệu và khả năng triển khai một sản phẩm web hoàn chỉnh.

### Mục đích nghiên cứu

*Về hệ thống:* Một website ghi chú cá nhân có các chức năng cơ bản: tạo, chỉnh sửa, xóa, lưu ghi chú với MySQL. Giao diện được thiết kế với HTML, CSS, JavaScript và Bootstrap Framework.

*Về mã nguồn:* Toàn bộ mã nguồn của dự án được quản lý trên Git, có cấu trúc rõ ràng. Ngoài ra, mã nguồn được chú thích cẩn thận, giúp dễ dàng cho việc đọc hiểu, kiểm thử và nâng cấp sau này.

*Về báo cáo kỹ thuật:* Quyền báo cáo chi tiết về quá trình phân tích, thiết kế, triển khai hệ thống và các kết quả đạt được.

Báo cáo kết quả kỹ thuật sẽ bao gồm các chương:

- Chương 1: Tổng quan nghiên cứu
- Chương 2: Nghiên cứu lý thuyết
- Chương 3: Hiện thực hóa nghiên cứu

- Chương 4: Kết quả nghiên cứu
- Chương 5: Kết luận và hướng phát triển

### Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống ghi chú cá nhân trực tuyến có khả năng quản lý người dùng, ghi chú, thẻ (tags) và lịch sử chỉnh sửa. Cụ thể:

**Người dùng (Users):** là đối tượng có thể đăng ký tài khoản, đăng nhập, đăng xuất và quản lý các ghi chú cá nhân. Mỗi người dùng có các thông tin cơ bản như tên đăng nhập, mật khẩu.

**Ghi chú (Notes):** là đối tượng chính của ứng dụng. Mỗi ghi chú có tiêu đề, nội dung, thời gian tạo và cập nhật. Ghi chú thuộc về người dùng cụ thể thông qua khóa ngoại user\_id. Ngoài ra, ghi chú có thể gắn với nhiều thẻ (tags) để phân loại và tìm kiếm dễ dàng.

**Thẻ (Tags):** cho phép người dùng gắn nhãn cho từng ghi chú, giúp việc tổ chức và phân loại nội dung hiệu quả hơn. Mỗi quan hệ giữa ghi chú và thẻ là N-N, được quản lý thông qua bảng trung gian note\_tags.

**Lịch sử chỉnh sửa (Note History):** lưu lại nội dung cũ mỗi khi người dùng cập nhật ghi chú. Việc này giúp người dùng theo dõi quá trình thay đổi và phục hồi thông tin nếu cần.

**Cơ sở dữ liệu MySQL:** là nơi lưu trữ tất cả dữ liệu liên quan đến người dùng, ghi chú, thẻ và lịch sử chỉnh sửa.

### Phương pháp nghiên cứu

*Phương pháp nghiên cứu lý thuyết:* Đề tài tập trung vào việc tìm hiểu và tổng hợp các kiến thức liên quan đến công nghệ web front-end và back-end như HTML, CSS, JavaScript, Bootstrap, PHP và MySQL. Ngoài ra còn tiến hành nghiên cứu mô hình hoạt động của website động, cơ chế xử lý yêu cầu và cách thức lưu trữ dữ liệu. Từ việc phân tích một số hệ thống ghi chú phổ biến như Google Keep, Evernote hay Notion, đề tài rút ra được những ưu điểm và hạn chế để định hướng cho việc xây dựng một hệ thống ghi chú cá nhân trực tuyến phù hợp. Việc nghiên cứu được chú



Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động  
trọng nhằm tạo ra một website giúp người dùng có trải nghiệm tốt, thân thiện và dễ sử dụng.

*Phương pháp nghiên cứu thực nghiệm:* Đề tài tiến hành xây dựng hệ thống ghi chú cá nhân trực tuyến, sau đó là giai đoạn thiết kế cơ sở dữ liệu bằng MySQL để lưu trữ thông tin người dùng và các ghi chú. Từ đó, tiến hành lập trình các chức năng chính của hệ thống như đăng ký, đăng nhập, tạo mới, chỉnh sửa, xóa, tìm kiếm và gắn thẻ ghi chú. Cuối cùng, tiến hành tổng hợp, phân tích kết quả thử nghiệm, ghi nhận các lỗi của hệ thống, rút ra bài học kinh nghiệm để từ đó có thể hoàn thiện hơn trong các phiên bản tiếp theo.

### **Phạm vi nghiên cứu**

Đề tài tập trung nghiên cứu và xây dựng một hệ thống ghi chú cá nhân trực tuyến có khả năng quản lý người dùng, ghi chú, thẻ (tags) và lịch sử chỉnh sửa. Hệ thống được phát triển với mục tiêu giúp người dùng có thể tạo, lưu trữ, tìm kiếm và chỉnh sửa các ghi chú của mình một cách dễ dàng và bảo mật.

Phạm vi bao gồm việc thiết kế và xây dựng website ghi chú cá nhân bằng các công nghệ Web Front-End như HTML, CSS, JavaScript kết hợp với PHP và MySQL. Ứng dụng được phát triển và chạy thử nghiệm trong môi trường XAMPP, có các chức năng chính bao gồm: đăng ký, đăng nhập, thêm - sửa - xóa ghi chú, gắn nhãn phân loại, tìm kiếm ghi chú và lưu lại lịch sử chỉnh sửa để người dùng có thể xem lại nội dung cũ.

## CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU

### 1.1 Tổng quan về vấn đề nghiên cứu

Trong thời đại công nghệ hiện nay, nhu cầu lưu trữ thông tin, ghi chú, theo dõi công việc và quản lý dữ liệu cá nhân ngày càng trở nên phổ biến. Người dùng hiện nay mong muốn có một công cụ ghi chú đơn giản, dễ sử dụng, có khả năng truy cập mọi lúc mọi nơi và tính bảo mật cao. Hiện nay đã xuất hiện nhiều ứng dụng ghi chú nổi bật như Google Keep, Evernote hay Notion. Tuy nhiên, mỗi ứng dụng đều có những hạn chế nhất định.

Để phục vụ nhu cầu học tập và nâng cao kỹ năng lập trình web, đề tài “Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động” được thực hiện nhằm xây dựng một hệ thống ghi chú hoạt động ổn định, giao diện thân thiện, có các chức năng cơ bản giúp người dùng có một trải nghiệm tốt.

### 1.2 Một số hệ thống ghi chú phổ biến

#### 1.2.1 Tổng quan về Google Keep

Google Keep là một dịch vụ ghi chú trực tuyến miễn phí do Google phát triển và được tích hợp trong hệ sinh thái Google Workspace (Gmail, Drive, Calendar,...). Ứng dụng ra mắt vào năm 2013, được thiết kế với sự nhanh nhẹn, nhẹ, đơn giản.

Đặc điểm: Hoạt động mượt trên mọi nền tảng, hỗ trợ nhiều loại ghi chú. Tuy nhiên, thiếu nhiều tính năng chuyên sâu, không có ứng dụng riêng cho máy tính.[1]



Hình 1.1 Google Keep

### 1.2.2 Phân tích Evernote

Evernote là một trong các ứng dụng ghi chú được ra đời từ năm 2008. Ứng dụng hướng đến các đối tượng như: nhân viên văn phòng, người quản lý dự án, sinh viên cần ghi chú chuyên sâu.

Đặc điểm: Hỗ trợ tài liệu dài, có phiên bản desktop, web, mobile. Tuy nhiên, bản miễn phí bị giới hạn, giá thành cao.[2]



Hình 1.2 Evernote

### 1.2.3 Giới thiệu về Notion

Notion ra mắt năm 2016, là một ứng dụng ghi chú thế hệ mới, kết hợp ghi chú, quản lý công việc, quản lý dự án.

Đặc điểm: Giao diện dạng khối, tạo cơ sở dữ liệu ngay trong ghi chú. Tuy nhiên, ứng dụng chưa hỗ trợ giao diện tiếng Việt, khó tiếp cận với người dùng chưa thành thạo tiếng Anh, cần nhiều thời gian để làm quen và sử dụng thành thục.[3]



Hình 1.3 Notion

## 1.3 Ưu điểm của hệ thống

Dựa trên việc phân tích các ứng dụng ở trên, có thể thấy rằng các ứng dụng trên vẫn có một số nhược điểm làm cho người dùng chưa có được trải nghiệm tốt nhất. Từ đó, đề tài hướng đến việc xây dựng một hệ thống ghi chú đơn giản, nhẹ và dễ sử dụng nhưng vẫn đáp ứng đầy đủ các chức năng cần thiết. Hệ thống được thiết kế theo các tiêu chí: giao diện thân thiện, cấu trúc quản lý rõ ràng, hỗ trợ lưu lịch sử

Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động  
chỉnh sửa, cơ sở dữ liệu được tổ chức khoa học và chuẩn hóa, từ đó mang lại trải  
nghiệm sử dụng thuận tiện và ổn định cho người dùng.

## CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

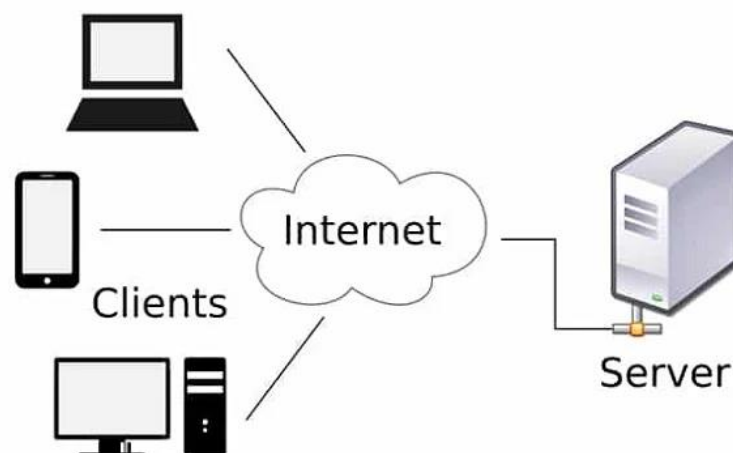
### 2.1 Tổng quan về công nghệ web

Web tĩnh (Static web) là trang web có nội dung cố định, được lưu dưới dạng HTML/CSS, nội dung chỉ thay đổi khi tệp thay đổi, thích hợp cho trang giới thiệu.

Web động (Dynamic web) là những trang web có nội dung trang được sinh ra tại thời điểm người dùng truy cập, dựa trên dữ liệu từ server trả về thông qua xử lý tương tác với cơ sở dữ liệu. Những trang web động cho phép tương tác với cơ sở dữ liệu để thực hiện các chức năng cơ bản như đăng ký, đăng nhập, quản lý dữ liệu,...

#### 2.1.1 Kiến trúc Client-Server

Kiến trúc Client-Server là kiến trúc nền tảng của hầu hết các ứng dụng web động hiện nay.



Hình 2.1 Mô hình Client–Server

Trong đó:

**Client (máy khách)** là thiết bị yêu cầu dịch vụ từ máy chủ. Máy khách sẽ gửi yêu cầu đến máy chủ, sau khi yêu cầu được xử lý, máy chủ sẽ gửi phản hồi đến máy khách.

**Server (máy chủ)** là máy tính thực hiện các nhiệm vụ xử lý yêu cầu, cung cấp các dịch vụ cho máy khách. Máy chủ luôn hoạt động và gửi phản hồi khi có yêu cầu từ máy khách.

Trong mô hình Client-Server, mọi sự tương tác giữa Client và Server đều diễn ra thông qua mạng máy tính. Môi trường kết nối bao gồm:

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

**Internet:** Môi trường phổ biến nhất, cho phép người dùng truy cập website ở bất kỳ nơi nào. Trình duyệt của client yêu cầu HTTP/HTTPS đến server thông qua đường truyền Internet.

**Mạng LAN (Local Area Network):** Được sử dụng trong các hệ thống nội bộ nhỏ hoặc ứng dụng doanh nghiệp. Máy khách và máy chủ giao tiếp trực tiếp trong cùng mạng nội bộ, giúp tốc độ truy cập nhanh và bảo mật dữ liệu tốt hơn.

**Localhost (môi trường máy tính cá nhân):** Thường được dùng trong lập trình và thử nghiệm, cả client và server cùng chạy trên một thiết bị, giao tiếp diễn ra thông qua địa chỉ đặc biệt là 127.0.0.1.

Mô hình Client–Server có các ưu điểm sau:

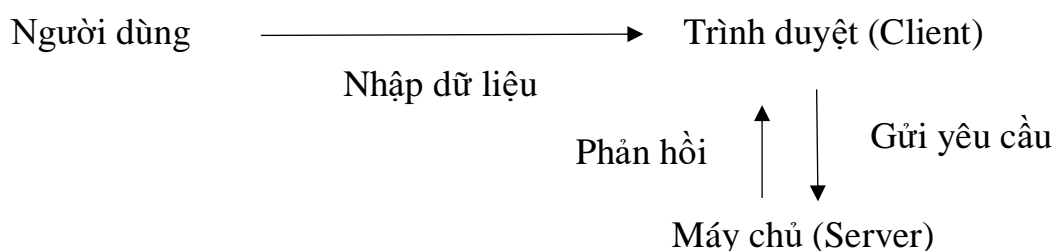
**Dễ quản lý:** Việc tập trung các thông tin và tài nguyên vào một vị trí duy nhất giúp cho việc quản lý, điều hành trở nên hiệu quả hơn. Quản trị viên có thể dễ dàng theo dõi và kiểm soát tài nguyên, cập nhật phần mềm

**Dễ mở rộng, bảo trì:** Mô hình này cho phép mở rộng các thành phần của hệ thống một cách linh hoạt.

**Khả năng truy cập:** Mọi client đều có thể đăng nhập vào hệ thống mạng máy tính, cho phép nhiều người truy cập cùng một lúc. Việc này giúp cho các nhân viên đều có thể truy cập thông tin của công ty một cách dễ dàng.

**Độ an toàn cao:** Các dữ liệu quan trọng như tài khoản, mật khẩu được lưu trữ an toàn tại bộ phận quản lý. Người dùng có thể kiểm soát truy cập, chỉ có ai được cấp quyền truy cập thì mới được thực hiện các thao tác trên các dữ liệu đó.

### 2.1.2 Quy trình hoạt động



Khi người dùng thực hiện thao tác như nhập ghi chú, xem danh sách hoặc chỉnh sửa dữ liệu, client sẽ tiếp nhận các thao tác này và chuyển chúng thành các yêu cầu HTTP (HTTP Request) gửi đến máy chủ. Các yêu cầu thường sử dụng phương thức GET hoặc POST tùy thuộc vào loại dữ liệu cần gửi. Server tiếp nhận yêu cầu, phân tích nội dung và xử lý logic nghiệp vụ. Đây là bước quan trọng nhằm quyết định hệ thống sẽ thực hiện hành động nào, chẳng hạn như lấy dữ liệu, lưu ghi chú mới hoặc cập nhật dữ liệu cũ.

Sau khi phân tích yêu cầu, server sẽ tương tác với hệ quản trị cơ sở dữ liệu MySQL để truy xuất hoặc cập nhật thông tin. Việc truy vấn được thực hiện thông qua các câu lệnh SQL như SELECT, INSERT, UPDATE hoặc DELETE. MySQL tiếp nhận truy vấn và trả về kết quả tương ứng cho server. Tất cả dữ liệu ghi chú, người dùng, thẻ tag hoặc lịch sử chỉnh sửa đều được lưu trữ có cấu trúc trong MySQL, đảm bảo khả năng quản lý và truy xuất nhanh chóng.

Khi đã có kết quả từ cơ sở dữ liệu, máy chủ sẽ xử lý dữ liệu và tạo ra phản hồi (HTTP Response). Trình duyệt nhận phản hồi và hiển thị thông tin trực quan cho người dùng thông qua giao diện web. Từ đó, người dùng tiếp tục thao tác, tạo thành một vòng lặp tương tác liên tục giữa client và server.

Nhìn chung, mô hình Client-Server tạo ra một quy trình xử lý tuần tự, rõ ràng và hiệu quả. Các thành phần được phân chia nhiệm vụ cụ thể: client đảm nhiệm giao tiếp với người dùng, server xử lý logic và MySQL chịu trách nhiệm lưu trữ dữ liệu. Mô hình này giúp hệ thống ghi chú cá nhân hoạt động ổn định, dễ mở rộng, dễ bảo trì và thân thiện với người dùng cuối.

## **2.2 Ngôn ngữ và công nghệ sử dụng**

Từ mô hình Client–Server đã trình bày trong 2.1, ta có thể thấy rằng cần xây dựng hệ thống ghi chú cá nhân trực tuyến tách biệt rõ ràng giữa phần giao diện người dùng và phần xử lý, lưu trữ dữ liệu. Client đảm bảo vai trò hiển thị thông tin và nhận thao tác từ người dùng, còn server thì xử lý logic nghiệp vụ, quản lý dữ liệu và đảm bảo tính bảo mật của hệ thống.

Để triển khai mô hình Client–Server một cách hiệu quả, cần lựa chọn những công nghệ phù hợp với các thành phần. Các công nghệ này đảm bảo khả năng xử lý linh hoạt, dễ phát triển, dễ bảo trì. Do đó, phần này sẽ trình bày chi tiết các công nghệ

Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động  
được sử dụng gồm có HTML, CSS, JavaScript cùng với PHP, MySQL, XAMPP và vai trò cụ thể của từng ngôn ngữ, công nghệ sử dụng.

### 2.2.1 HTML – Ngôn ngữ đánh dấu siêu văn bản

HTML (Hyper Text Markup Language) là ngôn ngữ đánh dấu chuẩn được sử dụng để xây dựng và cấu trúc các trang web trên Internet. HTML không phải là ngôn ngữ lập trình, mà có chức năng giống như phần mềm xử lý văn bản, được dùng để tạo bố cục, định dạng và hiển thị nội dung như tiêu đề, đoạn văn, hình ảnh và liên kết cho trang web.

*Vai trò và chức năng của HTML:*

**Xây dựng cấu trúc:** HTML sử dụng các thẻ để xác định các phần tử trong một trang web, bao gồm tiêu đề (<h1>), đoạn văn (<p>), danh sách (<ul>, <ol>), hình ảnh (<img>), và liên kết (<a>).

**Tạo siêu văn bản:** Nó cho phép tạo các liên kết (hyperlinks) cho phép người dùng điều hướng giữa các trang web khác nhau hoặc giữa các phần trên cùng một trang.

**Nhúng nội dung đa phương tiện:** HTML có thể nhúng hình ảnh, video, âm thanh và các đối tượng khác vào trang web để làm cho nội dung trở nên sinh động hơn, hấp dẫn hơn.

**Cơ sở cho các công nghệ khác:** HTML là nền tảng cơ bản cho website, và thường được kết hợp với CSS (tạo kiểu dáng) và giao diện và JavaScript (thêm chức năng tương tác động) để xây dựng một trang web hoàn chỉnh và tương tác.

### 2.2.2 CSS – Ngôn ngữ định dạng

CSS (Cascading Style Sheets) là một ngôn ngữ định dạng được sử dụng để tạo kiểu dáng và trình bày cho các trang web, bằng cách kiểm soát bố cục, màu sắc, phông chữ và các yếu tố hiển thị khác của nội dung được tạo bởi HTML. Về cơ bản, nó giúp tách biệt phần nội dung (HTML) và phần trình bày, làm cho mã nguồn gọn gàng hơn, dễ quản lý và dễ bảo trì hơn.

*Chức năng của CSS:* Giúp chúng ta định nghĩa cách hiển thị của các phần tử trên trang web. Ta có thể tùy chỉnh nhiều thuộc tính như:



**Màu sắc:** Màu chữ, màu nền, màu viền.

**Kiểu chữ:** Kiểu chữ, kích thước, độ dày, căn chỉnh.

**Bố cục:** Sắp xếp vị trí các phần tử, khoảng cách, lề.

**Hiệu ứng:** Tạo hiệu ứng hình ảnh, chuyển động, đổ bóng.

CSS có 3 loại là Inline style sheet (Bảng kiểu trực tiếp), Internal style sheet (Bảng kiểu được nhúng vào trong tài liệu HTML) và External style sheet (Bảng kiểu bên ngoài).

#### 2.2.2.1 *Inline style sheet*

Đây là kiểu được gán cho một dòng hoặc một đoạn văn bản, CSS được viết trực tiếp bên trong thẻ HTML thông qua thuộc tính style.

Cú pháp:

```
<TagName style= "property1: value 1; property2: value 2; ...; ">
```

Nội dung văn bản muốn nhận định dạng

```
</TagName>
```

#### 2.2.2.2 *Internal style sheet*

Là bảng mẫu thích hợp cho trang riêng lẻ với nhiều văn bản. Để tạo CSS loại này ta định nghĩa sẵn bảng mẫu chung trên phần đầu trang trong cặp thẻ <head> ... </head>. Ta áp dụng kiểu viết 1 cho CSS loại 2. Sau đó chỉ việc gọi các thẻ HTML trong <body> bình thường như đã biết về HTML. Tự trình duyệt sẽ cập nhật các định nghĩa thêm từ CSS vào cho các thẻ HTML đó.

Cú pháp:

```
<head>
```

```
<style type= "text/css">
```

```
Selector {property1: value1; property2: value2; ...; }
```

(Lặp lại cho mỗi thẻ có thuộc tính cần định dạng)

```
</style>
```

```
</head>
```

### 2.2.2.3 External style sheet

Là một bảng kiểu được lưu trữ thành một file bên ngoài và được liên kết với trang HTML. Do đó, ta có thể gọi chúng vào bất kỳ trang HTML nào ta muốn. Nếu đang sử dụng Dreamweaver, để soạn thảo tập tin CSS, ta vào File → New → Basic Page → CSS.

Cú pháp:

```
Selector {property1: value 1; property2: value 2; ...; }
```

Sau đó lưu tập tin lại, tập tin sẽ có phần mở rộng là \*.css. Để sử dụng bảng mẫu đã định nghĩa, ta gọi tập tin CSS trong phần <head> của trang với thẻ <link>.

### 2.2.3 JavaScript – Ngôn ngữ lập trình web

JavaScript là một ngôn ngữ lập trình đa năng, được sử dụng chủ yếu để tạo ra các trang web tương tác và động. Nó cho phép các nhà phát triển thêm các tính năng như hoạt ảnh, menu thả xuống, và bản đồ tương tác, giúp cải thiện trải nghiệm người dùng. Bên cạnh việc chạy trên trình duyệt, JavaScript còn được dùng để phát triển ứng dụng phía máy chủ, ứng dụng di động, và trò chơi.

**Chức năng của JavaScript:** Làm cho các trang web trở nên tương tác và sống động bằng cách thêm các yếu tố động, phản hồi người dùng và xử lý các hành động. Nó đóng vai trò thiết yếu trong việc thay đổi cấu trúc, nội dung và phong cách của trang web sau khi trang đã được tải lên trình duyệt, ngoài ra còn có khả năng phát triển ứng dụng phía máy chủ, di động, ...

Các chức năng chính trên trình duyệt:

**Tạo tương tác:** Cho phép tạo ra các tính năng như menu thả xuống, nút bấm có thể nhấp, biểu mẫu (form), slideshow hình ảnh và các hiệu ứng động.

**Thao tác với DOM:** Thêm, xóa, sửa đổi các phần tử HTML và thuộc tính CSS của trang web một cách linh hoạt để thay đổi bố cục và nội dung.

**Xác nhận dữ liệu:** Kiểm tra tính hợp lệ của dữ liệu người dùng nhập vào biểu mẫu trước khi gửi đi.

**Giao tiếp với máy chủ:** Gửi yêu cầu đến máy chủ và nhận phản hồi để cập nhật một phần nội dung trang mà không cần tải lại toàn bộ trang.

#### 2.2.4 Mối quan hệ giữa HTML – CSS – JavaScript

Ba công nghệ HTML, CSS và JavaScript là ba trụ cột chính của lập trình Web Front-End. Chúng kết hợp với nhau để tạo nên một trang web hoàn chỉnh và tương tác tốt với người dùng.

**HTML (Hyper Text Markup Language):** Là ngôn ngữ đánh dấu giúp định nghĩa cấu trúc và nội dung cho trang web. nơi chứa các phần tử như tiêu đề, đoạn văn, hình ảnh, liên kết,...

**CSS (Cascading Style Sheets):** Được sử dụng để trang trí và định dạng cho HTML. CSS quyết định cách mà nội dung hiển thị, bao gồm màu sắc, phông chữ, bố cục, khoảng cách,... Giúp trang web trở nên trực quan, đẹp mắt và dễ sử dụng.

**JavaScript:** Là ngôn ngữ lập trình dùng để xử lý tương tác và thêm tính năng động cho website. JavaScript có thể lắng nghe sự kiện (click, nhập liệu), thay đổi nội dung HTML, thao tác dữ liệu và tạo các hiệu ứng trực quan.

#### 2.2.5 Tổng quan về Bootstrap

Bootstrap là một framework front-end mã nguồn mở nhằm hỗ trợ lập trình viên xây dựng giao diện web hiện đại, thống nhất và dễ tùy biến. Framework này cung cấp sẵn các tệp CSS và JavaScript, cùng hàng trăm class giúp tạo bố cục, định dạng văn bản, thiết kế biểu mẫu và các thành phần giao diện mà không cần tự viết CSS từ đầu. Với hệ thống lưới 12 cột (Grid System), Bootstrap cho phép trang web tự điều chỉnh kích thước và bố cục tương ứng với điện thoại, máy tính bảng hoặc máy tính để bàn, giúp website đạt tính responsive cao.

Bootstrap hoạt động như một bộ công cụ giao diện hoàn chỉnh với nhiều thành phần dựng sẵn, vì vậy nó giúp đơn giản hóa quá trình thiết kế web và giảm thiểu lỗi. Bên cạnh CSS, Bootstrap còn cung cấp thư viện JavaScript nhằm xử lý các thành phần tương tác như Dropdown, Collapse, Carousel... giúp website trở nên sinh động và tiện dụng hơn. Nhờ đó, lập trình viên có thể xây dựng giao diện nhanh chóng mà vẫn đảm bảo tính chuyên nghiệp, thẩm mỹ và đồng nhất.

Một số câu lệnh trong Bootstrap:

### 2.2.5.1 Bootstrap Containers

Class `.container` cung cấp một vùng chứa có chiều rộng cố định và class `.container-fluid` cung cấp một vùng chứa có chiều rộng đầy đủ, trải dài toàn bộ chiều rộng của khung. Ví dụ:

```
<div class="container">
  <h1>Ghi chú cá nhân</h1>
</div>
```

```
<div class="container-fluid">
  <h1>Ghi chú cá nhân</h1>
</div>
```

### 2.2.5.2 Bootstrap Buttons

Bootstrap cung cấp nhiều kiểu nút khác nhau như: `primary`, `secondary`, `success`, `info`,... Ví dụ:

```
<button type = "button" class = "btn btn-primary">
DA23TTD </button>

<button type = "button" class = "btn btn-secondary">
DA23TTD </button>

<button type = "button" class = "btn btn-success">
DA23TTD </button>

<button type = "button" class = "btn btn-info"> DA23TTD
</button>
```

Các lệnh trên sẽ tạo các nút có màu xanh dương, màu xám, màu xanh lá và màu xanh biển.

### 2.2.5.3 Bootstrap Alerts

Bootstrap cung cấp một cách dễ dàng để tạo các thông báo cảnh báo được xác định trước. Cảnh báo được tạo bằng class .alert , theo sau là một trong các lớp theo ngữ cảnh: .alert-success, .alert-info, .alert-warning,... Ví dụ:

```
<div class = "alert alert-success">

    DA23TTD

</div>
```

### 2.2.5.4 Bootstrap Tables

Bảng bootstrap cơ bản có phần đệm nhẹ và các ngăn chia theo chiều ngang. Class .table thêm kiểu dáng cơ bản vào bảng. Ví dụ:

```
<div class="container mt-3">

    <table class="table">

        <thead>

            <tr>

                <th>Firstname</th>

                <th>Lastname</th>

                <th>Email</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td>Trương</td>

                <td>Minh</td>

                <td>110123263@st.tvu.edu.vn</td>

            </tr>

            <tr>
```

```
<td>Trần</td>

<td>Tín</td>

<td>110123186@st.tvu.edu.vn</td>

</tr>

</tbody>

</table>

</div>
```

#### 2.2.5.5 Bootstrap Navbar

Với Bootstrap, thanh điều hướng có thể mở rộng hoặc thu gọn tùy thuộc vào kích thước màn hình. Thanh điều hướng được tạo bằng class .navbar. Ví dụ:

```
<nav class="navbar navbar-expand-sm bg-light">

  <div class="container">

    <ul class="navbar-nav">

      <li class="nav-item">

        <a class="nav-link" href="#">Trang chủ</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="#">Liên hệ</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="#">Sản phẩm</a>

      </li>

    </ul>

  </div>

</nav>
```

### 2.2.5.6 Bootstrap Forms

Các phần tử `<input>` và `<textarea>` dạng văn bản có class `.form-control` có kiểu dáng biểu mẫu phù hợp. Ví dụ:

```
<form method = "post" action = "">

  <div class = "mb-4">

    <label class="form-label"> Tên đăng nhập: </label>

    <input type = "text" name = "username" class =
"form-control" value = "" required>

  </div>

  <div class = "mb-4">

    <label class="form-label"> Mật khẩu: </label>

    <input type = "password" name = "password" class =
"form-control" value = "" required>

  </div>

  <button type = "submit" class = "btn btn-primary">
Đăng nhập </button>

</form>
```

### 2.2.5.7 Bootstrap Grid

Hệ thống lưới Bootstrap có sáu lớp:

**.col** (thiết bị cực nhỏ - chiều rộng màn hình nhỏ hơn 576px)

**.col-sm-** (thiết bị nhỏ - chiều rộng màn hình bằng hoặc lớn hơn 576px)

**.col-md-** (thiết bị trung bình - chiều rộng màn hình bằng hoặc lớn hơn 768px)

**.col-lg-** (thiết bị lớn - chiều rộng màn hình bằng hoặc lớn hơn 992px)

**.col-xl-** (thiết bị cực lớn - chiều rộng màn hình bằng hoặc lớn hơn 1200px)

**.col-xxl-** (thiết bị xxlarge - chiều rộng màn hình bằng hoặc lớn hơn 1400px)

Ví dụ:

```
<div class="row">

    <div class="col-sm-6 bg-primary text-white"> 50%
</div>

    <div class="col-sm-6 bg-dark text-white"> 50% </div>

</div>
```

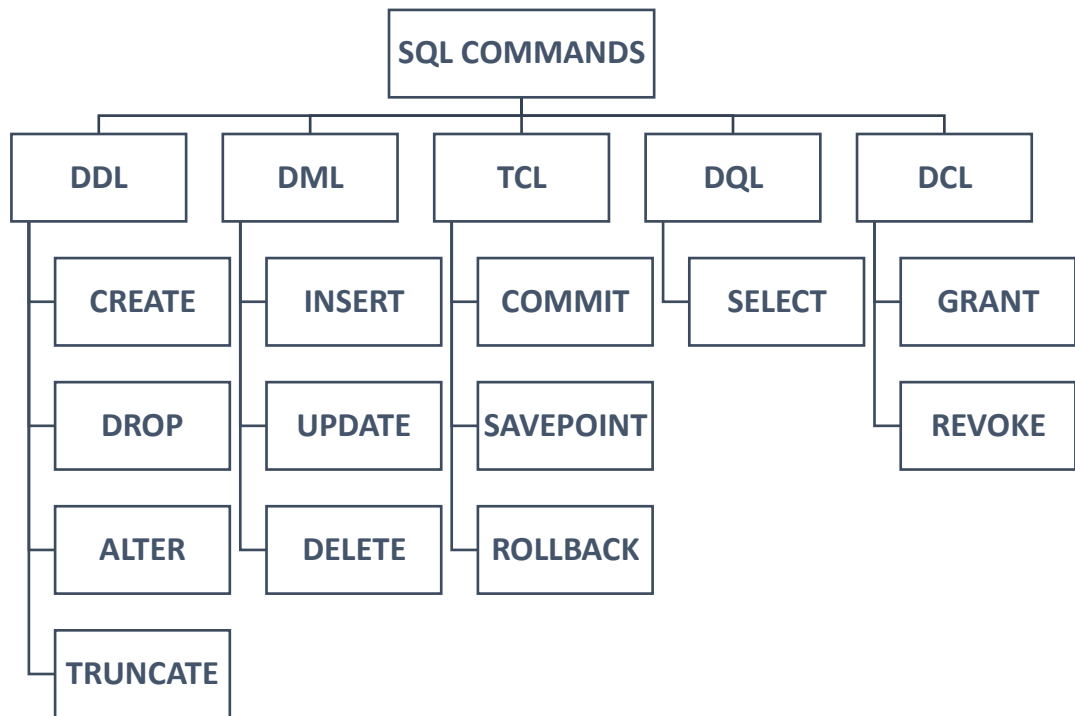
### **2.2.6 MySQL – Hệ quản trị cơ sở dữ liệu**

Trong số các hệ quản trị cơ sở dữ liệu quan hệ hiện nay như PostgreSQL, SQL Server hay Oracle, đồ án lựa chọn MySQL bởi đây là hệ quản trị cơ sở dữ liệu phù hợp nhất với quy mô, mục tiêu và phạm vi của hệ thống ghi chú cá nhân trực tuyến. MySQL có đặc điểm là mã nguồn mở, dễ dàng cài đặt và sử dụng, được tích hợp sẵn trong XAMPP, giúp dễ dàng triển khai nhanh chóng mà không cần cấu hình phức tạp. Ngoài ra, MySQL có tốc độ xử lý truy vấn nhanh, hoạt động ổn định và hỗ trợ tốt cho các thao tác CRUD – những thao tác cốt lõi trong hệ thống ghi chú.

MySQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở được sử dụng rộng rãi trong lập trình web. MySQL cho phép người dùng tạo, lưu trữ, truy xuất, sửa đổi và quản lý dữ liệu một cách hiệu quả bằng cách sử dụng ngôn ngữ SQL. Đây là ngôn ngữ tiêu chuẩn để thao tác với cơ sở dữ liệu quan hệ.

Các lệnh SQL được phân thành năm loại là DDL – Ngôn ngữ định nghĩa dữ liệu, DML – Ngôn ngữ truy vấn dữ liệu, TCL – Ngôn ngữ thao tác dữ liệu, DQL – Ngôn ngữ điều khiển dữ liệu và DCL – Ngôn ngữ điều khiển giao dịch.





Hình 2.2 Phân loại lệnh SQL

DDL bao gồm các lệnh SQL được sử dụng để định nghĩa, thay đổi và xóa các cấu trúc cơ sở dữ liệu như bảng. Nó chỉ đơn giản xử lý các mô tả về lược đồ cơ sở dữ liệu và được sử dụng để tạo và sửa đổi cấu trúc của các đối tượng cơ sở dữ liệu trong cơ sở dữ liệu.

Bảng 2.1 Các lệnh SQL DDL

Yêu cầu	Miêu tả	Cú pháp
CREATE	Tạo cơ sở dữ liệu hoặc các đối tượng.	CREATE TABLE table_name (column1 data_type, column2 data_type, ...);
DROP	Xóa các đối tượng khỏi cơ sở dữ liệu.	DROP TABLE table_name;
ALTER	Thay đổi cấu trúc của cơ sở dữ liệu.	ALTER TABLE table_name ADD COLUMN column_name data_type;
TRUNCATE	Xóa tất cả bản ghi khỏi bảng.	TRUNCATE TABLE table_name;

Ví dụ:

```
CREATE TABLE notes (  
id INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
username VARCHAR(50) NOT NULL,  
password VARCHAR(50) NOT NULL,  
created_at DATETIME  
)
```

DQL được sử dụng để lấy dữ liệu từ cơ sở dữ liệu. Lệnh chính là SELECT, lệnh này truy xuất các bản ghi dựa trên truy vấn. Kết quả được trả về dưới dạng một tập kết quả (một bảng tạm thời) có thể được xem hoặc sử dụng trong các ứng dụng.

Bảng 2.2 Các lệnh SQL DQL

Yêu cầu	Miêu tả	Cú pháp
SELECT	Lấy dữ liệu từ cơ sở dữ liệu.	SELECT column1, column2, ...FROM table_name WHERE condition;
FROM	Chỉ ra các bảng để lấy dữ liệu.	SELECT column1 FROM table_name;
WHERE	Lọc các hàng trước khi nhóm hoặc tổng hợp.	SELECT column1 FROM table_name WHERE condition;

Ví dụ :

```
SELECT user_id, title, content FROM notes WHERE user_id  
= $user_id;
```

DML được sử dụng để thao tác dữ liệu được lưu trữ trong các bảng cơ sở dữ liệu. Với DML, ta có thể chèn bản ghi mới, cập nhật bản ghi hiện có, xóa dữ liệu hoặc truy xuất thông tin.

Bảng 2.3 Các lệnh SQL DML

Yêu cầu	Miêu tả	Cú pháp
INSERT	Chèn dữ liệu vào bảng.	INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);
UPDATE	Cập nhật dữ liệu hiện có trong bảng.	UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;
DELETE	Xóa bản ghi khỏi bảng cơ sở dữ liệu.	DELETE FROM table_name WHERE condition;

Ví dụ :

```
INSERT INTO notes (user_id, title, content) VALUES
($user_id, '$title', '$content');
```

DCL bao gồm các lệnh như GRANT và REVOKE, chủ yếu xử lý các quyền, phân quyền và các điều khiển khác của hệ thống cơ sở dữ liệu. Các lệnh này được sử dụng để kiểm soát quyền truy cập dữ liệu trong cơ sở dữ liệu bằng cách cấp hoặc thu hồi quyền.

Bảng 2.4 Các lệnh SQL DCL

Yêu cầu	Miêu tả	Cú pháp
GRANT	Chỉ định các quyền mới cho tài khoản của người dùng, cho phép truy cập vào các đối tượng cơ sở dữ liệu cụ thể.	GRANT privilege_type [(column_list)] ON [object_type] object_name TO user [WITH GRANT OPTION];
REVOKE	Xóa các đặc quyền đã cấp trước đó khỏi tài khoản của người dùng, tước quyền truy cập vào các đối tượng cơ sở dữ liệu.	REVOKE [GRANT OPTION FOR] privilege_type [(column_list)] ON [object_type] object_name FROM user [CASCADE];

Ví dụ :

```
GRANT SELECT, INSERT ON note_db.notes TO  
'note_user'@'localhost';
```

TCL giao dịch nhóm một tập hợp các tác vụ thành một đơn vị thực thi duy nhất. Mỗi giao dịch bắt đầu bằng một tác vụ cụ thể và kết thúc khi tất cả các tác vụ trong nhóm được hoàn thành thành công. Nếu bất kỳ tác vụ nào thất bại, giao dịch sẽ thất bại. Do đó, một giao dịch chỉ có hai kết quả: thành công hoặc thất bại.

Bảng 2.5 Các lệnh SQL TCL

Yêu cầu	Miêu tả	Cú pháp
COMMIT	Lưu tất cả các thay đổi được thực hiện trong quá trình giao dịch	COMMIT;
SAVEPOINT	Tạo một điểm lưu trong giao dịch hiện tại	SAVEPOINT savepoint_name;
ROLLBACK	Hoàn tác các thay đổi được thực hiện trong quá trình giao dịch	ROLLBACK;
BEGIN TRANSACTION	Bắt đầu một giao dịch mới	BEGIN TRANSACTION [transaction_name];

Ví dụ:

```
BEGIN TRANSACTION;  
  
INSERT INTO notes (user_id, title, content)  
VALUES (1, 'Học tập', 'Học bài lúc 19h');  
  
COMMIT;
```

Một số câu lệnh của MySQL:

#### 2.2.6.1 MySQL Create Table

Câu lệnh CREATE TABLE được sử dụng để tạo bảng trong MySQL.

Cú pháp:

```
CREATE TABLE notes (  
  id INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL,  
  password VARCHAR(50) NOT NULL,  
  created_at DATETIME  
)
```

#### 2.2.6.2 MySQL Insert Data

Câu lệnh INSERT INTO được sử dụng để thêm bản ghi mới vào bảng MySQL.

Cú pháp:

```
INSERT INTO table_name (column1, column2, column3,...)  
VALUE (value1, value2, value3,...)
```

#### 2.2.6.3 MySQL Select Data

Câu lệnh SELECT được sử dụng để chọn dữ liệu từ một hoặc nhiều bảng.

Cú pháp:

```
SELECT column_name(s) FROM table_name
```

Sử dụng ký tự \* để chọn tất cả các cột từ một bảng.

```
SELECT * FROM table_name
```

#### 2.2.6.4 MySQL Where

Mệnh đề WHERE được sử dụng để lọc bản ghi, chỉ trích xuất những bản ghi đáp ứng một điều kiện cụ thể.

Cú pháp:

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

```
SELECT column_name(s) FROM table_name WHERE column_name  
operator value
```

### 2.2.6.5 MySQL Order By

Mệnh đề ORDER BY được sử dụng để sắp xếp tập kết quả theo thứ tự tăng dần hoặc giảm dần.

Mệnh đề ORDER BY mặc định sắp xếp các bản ghi theo thứ tự tăng dần. Để sắp xếp các bản ghi theo thứ tự giảm dần, ta sử dụng từ khóa DESC.

Cú pháp:

```
SELECT column_name(s) FROM table_name ORDER BY  
column_name(s) operator value ASC|DESC
```

### 2.2.6.6 MySQL Delete Data

Câu lệnh DELETE được sử dụng để xóa các bản ghi khỏi một bảng.

Cú pháp:

```
DELETE FROM table_name  
  
WHERE some_column = some_value
```

Mệnh đề WHERE chỉ định bản ghi hoặc các bản ghi nào cần xóa. Nếu bỏ qua mệnh đề WHERE, tất cả bản ghi sẽ bị xóa!

### 2.2.6.7 MySQL Update Data

Câu lệnh UPDATE được sử dụng để cập nhật các bản ghi hiện có trong một bảng.

Cú pháp:

```
UPDATE table_name  
  
SET column1 = value, column2 = value2,...  
  
WHERE some_column = some_value
```

Mệnh đề WHERE chỉ định bản ghi hoặc các bản ghi nào cần được cập nhật. Nếu bạn bỏ qua mệnh đề WHERE, tất cả các bản ghi sẽ được cập nhật!

## 2.2.7 PHP – Công nghệ xử lý dữ liệu và phát triển web động

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản phía máy chủ được thiết kế chủ yếu để phát triển các ứng dụng web động. PHP có thể nhúng trực tiếp vào mã HTML, giúp dễ dàng tạo ra các trang web có khả năng tương tác, truy xuất và xử lý dữ liệu.

PHP có thể xử lý dữ liệu động như đăng nhập, lưu trữ thông tin, đọc và ghi dữ liệu từ cơ sở dữ liệu. Mỗi khi người dùng truy cập một trang PHP, mã PHP sẽ được máy chủ xử lý trước, sau đó kết quả được gửi đến trình duyệt.

Cú pháp:

```
<?php  
  
// Code PHP  
  
?>
```

### 2.2.7.1 Biến trong PHP

Biến trong PHP luôn bắt đầu bằng ký tự \$, không có lệnh khai báo biến. Biến được tạo ngay khi ta gán giá trị cho nó lần đầu tiên. Ví dụ:

```
$name = "Minh";  
  
$class = "DA23TTD";
```

Quy tắc cho biến PHP:

1. Một biến bắt đầu bằng \$dấu, theo sau là tên của biến.
2. Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới.
3. Tên biến không thể bắt đầu bằng số.
4. Tên biến chỉ chứa các ký tự chữ và số và dấu gạch dưới (Az, 0-9 và \_).
5. Tên biến phân biệt chữ hoa và chữ thường (\$name và \$NAME là hai biến khác nhau).

### 2.2.7.2 Câu lệnh if

Trong PHP chúng ta có các câu lệnh điều kiện sau:

1. if - thực thi câu lệnh nếu đúng với điều kiện.

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

2. if...else - thực thi câu lệnh nếu đúng với điều kiện và ngược lại nếu sai với điều kiện.

3. if...else if...else - thực thi câu lệnh với hai điều kiện.

Cú pháp 1:

```
if (điều kiện) {  
    // Khối lệnh;  
}
```

Cú pháp này dùng cho trường hợp chỉ cần kiểm tra điều kiện, nếu thỏa điều kiện thì thực hiện khối lệnh. Nếu không thỏa thì không làm gì cả.

Cú pháp 2:

```
if (biểu thức điều kiện) {  
    // Khối lệnh 1;  
} else {  
    // Khối lệnh 2;  
}
```

Cú pháp này dùng cho trường hợp nếu thỏa điều kiện thì thực hiện khối lệnh 1. Ngược lại nếu không thỏa thì thực hiện khối lệnh 2.

Cú pháp 3:

```
if (biểu thức điều kiện 1) {  
    // Khối lệnh 1;  
} else {  
    if (biểu thức điều kiện 2) {  
        // Khối lệnh 2;  
    } else {  
        // Khối lệnh 3;  
    }  
}
```



```
}
```

Cú pháp này được dùng nếu có nhiều hơn 2 điều kiện.

#### 2.2.7.3 Câu lệnh vòng lặp *for*

Câu lệnh *for* được dùng để lặp qua một khối lệnh với số lần xác định.

Cú pháp:

```
for (biểu thức 1, biểu thức 2, biểu thức 3) {  
    // code  
}
```

#### 2.2.7.4 Câu lệnh vòng lặp *while*

Câu lệnh *While* được dùng để lặp qua một khối lệnh nếu đúng với điều kiện.

Cú pháp:

```
while (biểu thức điều kiện) {  
    //code  
}
```

#### 2.2.7.5 Câu lệnh vòng lặp *do...while*

Câu lệnh *do ... while* sẽ luôn thực thi khối lệnh ít nhất một lần ngay cả khi điều kiện sai, sau đó kiểm tra điều kiện và lặp lại vòng lặp cho đến khi nào sai với điều kiện.

Cú pháp:

```
do {  
    //code  
} while (biểu thức điều kiện);
```

#### 2.2.7.6 Hàm trong *PHP*

Hàm được khai báo do người dùng định nghĩa bắt đầu bằng *function*, sau đó là tên hàm. Tên hàm phải bắt đầu bằng một chữ cái hoặc dấu gạch dưới, không phân biệt chữ hoa chữ thường. Ví dụ:

```
function class() {  
    echo "Lop DA23TTD";  
}  
  
class();
```

#### 2.2.7.7 Lập trình hướng đối tượng PHP – PHP OOP

Lập trình hướng đối tượng là tạo ra các đối tượng chứa cả dữ liệu và hàm.

##### **Class/Object:**

Lớp là khuôn mẫu cho các đối tượng và đối tượng là một thể hiện của lớp.

Cú pháp:

```
<?php  
  
class Name {  
    //code  
}  
?>
```

##### **Constructor:**

Nếu ta tạo một hàm `__construct()`, PHP sẽ tự động gọi hàm này khi ta tạo một đối tượng từ một lớp. Hàm xây dựng bắt đầu bằng hai dấu gạch dưới (`__`). Ví dụ:

```
<?php  
  
class Drink{  
    public $name;  
  
    function __construct($name) {  
        $this->name = $name;  
    }  
  
    function get_name() {  
        return $this->name;  
    }  
}
```

```
}  
  
}  
  
$pepsi = new Drink("Pepsi");  
echo $pepsi->get_name();  
  
?>
```

### Phương thức:

Phương thức tĩnh có thể được gọi trực tiếp mà không cần tạo phiên bản nào của lớp trước, được khai báo bằng từ khóa static. Để truy cập phương thức tĩnh, ta sử dụng tên lớp, dấu hai chấm kép (::) và tên phương thức.

Cú pháp:

```
<?php  
  
class Classname() {  
  
    public static function staticMethod() {  
  
        echo "Hello World!";  
  
    }  
  
}  
  
Classname::staticMethod();  
  
?>
```

### Thuộc tính:

Thuộc tính tĩnh có thể được gọi trực tiếp mà không cần tạo ra một thể hiện của lớp, được khai báo bằng từ khóa static. Để truy cập thuộc tính tĩnh, ta sử dụng tên lớp, dấu hai chấm kép (::) và tên thuộc tính.

Cú pháp:

```
<?php
```

```
class Classname {  
  
    public static $staticProp = "W3School";  
  
}  
  
Classname::$staticProp;  
  
?>
```

### 2.2.8 Quản lý dữ liệu với PHP và MySQL

Trong web động, việc lưu trữ và quản lý dữ liệu đóng vai trò quan trọng để đảm bảo hệ thống hoạt động ổn định và có thể mở rộng. PHP được dùng để xử lý logic phía máy chủ và MySQL đảm nhiệm vai trò lưu trữ dữ liệu. Hai công nghệ này thường đi chung với nhau trong các ứng dụng web quy mô nhỏ đến trung bình, đặc biệt trong môi trường học tập và thực hành.

Để thuận tiện trong quá trình phát triển, đồ án sử dụng XAMPP – phần mềm tích hợp gồm Apache, PHP và MySQL, việc này giúp chạy ứng dụng trên máy cá nhân mà không cần cấu hình thủ công. XAMPP cũng cung cấp phpMyAdmin, cho phép quản lý cơ sở dữ liệu trực quan, dễ thao tác.

Trong mục này, chúng ta sẽ tìm hiểu cách PHP kết nối với MySQL cũng như thực hiện các thao tác cơ bản như chọn, thêm, sửa, xóa dữ liệu – những thao tác cốt lõi để xây dựng hệ thống ghi chú trực tuyến.

#### 2.2.8.1 Kết nối csdl

Kết nối PHP-MySQL thường được thực hiện bằng thư viện MySQLi hoặc PDO. Đồ án sử dụng MySQLi vì cú pháp đơn giản và phù hợp với phạm vi môn học.

Ví dụ đoạn mã kết nối:

```
<?php  
  
$servername = "localhost";  
  
$username   = "root";  
  
$password   = "";  
  
$database   = "notes_db";
```

```
// Create connection

$conn = new mysqli($servername, $username, $password,
    $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$conn->set_charset("utf8mb4");
include_once("connect.php");
?>
```

#### 2.2.8.2 Các thao tác trên csdl

Các thao tác sau đây được sử dụng trong hệ thống: chọn, thêm, chỉnh sửa, xóa ghi chú.

##### a) Chọn dữ liệu (SELECT)

Dùng để lấy danh sách ghi chú, lịch sử chỉnh sửa hoặc nhãn dán của người dùng. Ví dụ:

```
$sql = "SELECT id, username, password FROM users WHERE
    username = '$username'";
```

##### b) Thêm dữ liệu (INSERT)

Dùng khi người dùng tạo ghi chú mới hoặc thêm nhãn. Ví dụ:

```
$sql = "INSERT INTO notes (user_id, title, content)
    VALUES ($user_id, '$title', '$content')";
```

##### c) Chỉnh sửa dữ liệu (UPDATE)

Dùng khi người dùng chỉnh sửa nội dung ghi chú. Ví dụ:

```
$sql = "UPDATE notes SET title='$title',  
content='$content' WHERE id=$note_id";
```

#### d) Xóa dữ liệu tạm thời

Dùng để xóa dữ liệu tạm thời, không xóa hẳn khỏi bảng. Ví dụ:

```
$sql = "UPDATE notes SET is_deleted = 1 WHERE id =  
$note_id AND user_id = $user_id";
```

### 2.3 Tổng kết chương

Chương 2 đã trình bày những nền tảng lý thuyết và công nghệ cốt lõi phục vụ cho việc xây dựng hệ thống ghi chú cá nhân trực tuyến, tập trung làm rõ kiến thức về web tĩnh – web động và mô hình Client–Server, qua đó giải thích cách thức client gửi yêu cầu, server xử lý và truy xuất dữ liệu từ MySQL. Ở phần Front-End, các công nghệ HTML, CSS, JavaScript và Bootstrap được giới thiệu nhằm đảm bảo giao diện thân thiện, dễ sử dụng và tương thích đa nền tảng. Phần Back-End bao gồm PHP – ngôn ngữ xử lý logic và tạo nội dung động, cùng với MySQL – hệ quản trị cơ sở dữ liệu phù hợp cho các thao tác CRUD. XAMPP được sử dụng làm môi trường phát triển tích hợp, hỗ trợ chạy Apache, PHP và MySQL trên máy cá nhân. Nhìn chung, toàn bộ kiến thức trong chương là cơ sở quan trọng để triển khai thiết kế cơ sở dữ liệu và xây dựng chức năng hệ thống ở Chương 3.

## CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1 Mô tả bài toán

Trong thời đại công nghệ số hiện nay, việc ghi chú, lưu trữ và quản lý thông tin cá nhân là nhu cầu thiết yếu của hầu hết mọi người. Tuy nhiên, những hình thức ghi chú thủ công hoặc ứng dụng ghi chú cục bộ trên thiết bị cá nhân thường gặp hạn chế trong việc truy cập từ nhiều thiết bị.

Đề tài “Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động” nhằm mục đích giải quyết vấn đề trên bằng cách xây dựng một website cho phép người dùng quản lý ghi chú trực tuyến mọi lúc, mọi nơi.

Người dùng có thể đăng ký, đăng nhập tài khoản, tạo, thêm, chỉnh sửa, xóa, gắn nhãn và tìm kiếm ghi chú của mình. Ngoài ra, hệ thống còn hỗ trợ lưu lại lịch sử chỉnh sửa, giúp người dùng có thể khôi phục nội dung cũ khi cần thiết.

Bài toán được xây dựng dựa trên mô hình Client–Server, trong đó:

**Client** là giao diện người dùng được thiết kế bằng HTML, CSS, JavaScript, Bootstrap.

**Server** xử lý logic bằng PHP.

**MySQL** dùng để lưu trữ thông tin người dùng, ghi chú, nhãn và lịch sử chỉnh sửa.

Hệ thống hướng đến việc tạo ra một ứng dụng web động dễ sử dụng, giao diện thân thiện, hoạt động ổn định trong môi trường XAMPP.

### 3.2 Đặc tả yêu cầu hệ thống

Để đảm bảo hệ thống ghi chú cá nhân trực tuyến được xây dựng một cách đầy đủ, chính xác và phù hợp với mục tiêu đề ra, đặc tả yêu cầu hệ thống là bước quan trọng nhằm xác định rõ những gì hệ thống cần phải thực hiện. Các yêu cầu này được chia thành hai nhóm: yêu cầu chức năng – mô tả những hoạt động mà hệ thống phải cung cấp, và yêu cầu phi chức năng – mô tả các tiêu chí về hiệu năng, bảo mật, độ tin cậy và khả năng sử dụng. Việc phân tích và đặc tả yêu cầu giúp đảm bảo quá trình thiết kế, triển khai và kiểm thử được thống nhất và có định hướng rõ ràng.

### 3.2.1 Yêu cầu chức năng

Bảng 3.1 Các yêu cầu chức năng

STT	Tên yêu cầu	Mô tả	Quyền	
			Người quản trị	Người dùng
1	Quản lý tài khoản người dùng	Hệ thống cho phép người dùng đăng ký, đăng nhập, đăng xuất tài khoản.	<input checked="" type="checkbox"/>	
2	Tạo ghi chú mới	Người dùng có thể tạo ghi chú gồm tiêu đề và nội dung.		<input checked="" type="checkbox"/>
3	Chỉnh sửa ghi chú	Người dùng chỉnh sửa nội dung ghi chú đã tạo và hệ thống tự động lưu lại lịch sử chỉnh sửa.		<input checked="" type="checkbox"/>
4	Xóa tạm thời	Ghi chú không bị xóa hoàn toàn mà có thể phục hồi khi cần.		<input checked="" type="checkbox"/>
5	Quản lý nhãn	Người dùng tạo, sửa tên nhãn và gán nhãn vào từng ghi chú.		<input checked="" type="checkbox"/>
6	Lọc ghi chú theo nhãn	Người dùng có thể lọc danh sách theo từng nhãn.		<input checked="" type="checkbox"/>
7	Tìm kiếm ghi chú	Hệ thống hỗ trợ tìm nhanh ghi chú.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Hiển thị lịch sử chỉnh sửa	Hiển thị danh sách các ghi chú trước khi người dùng chỉnh sửa.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Hiển thị giao diện	Hiển thị danh sách ghi chú (sắp xếp theo thời gian cập nhật gần nhất) và hỗ trợ hiển thị/lọc theo nhãn tương ứng.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### 3.2.2 Yêu cầu phi chức năng

Bảng 3.2 Các yêu cầu phi chức năng

STT	Tên yêu cầu	Mô tả
-----	-------------	-------



## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

1	Bảo mật	Mật khẩu được mã hóa.
2	Tính ổn định	Hệ thống hoạt động ổn định, hạn chế lỗi khi thao tác.
3	Hiệu năng	Các thao tác được xử lý nhanh gọn, phản hồi sau vài giây.
4	Tương thích	Hoạt động tốt trên các trình duyệt phổ biến như Chrome, Firefox, Cốc Cốc.

### 3.3 Thiết kế dữ liệu

Trong đồ án, cơ sở dữ liệu đóng vai trò quan trọng lưu trữ toàn bộ thông tin mà người dùng tạo ra và thao tác. Thiết kế dữ liệu là bước mô tả cách hệ thống tổ chức và lưu trữ thông tin phục vụ cho website ghi chú trực tuyến, với cơ sở dữ liệu được xây dựng theo mô hình quan hệ và triển khai trên MySQL, đảm bảo tính toàn vẹn, dễ truy vấn và thuận lợi khi mở rộng.

Hệ thống được thiết kế theo mô hình cơ sở dữ liệu quan hệ với hệ quản trị MySQL. Các dữ liệu được chia thành 5 bảng gồm: users, notes, tags, note\_tags, note\_history. Mỗi bảng đều có khóa chính (PK) để đảm bảo tính duy nhất và một số bảng có khóa ngoại (FK) nhằm liên kết các bảng với nhau, tạo mối quan hệ giữa các thực thể.

Việc thiết kế này giúp cho hệ thống không dư thừa dữ liệu, các bảng được tách riêng rõ ràng nhiệm vụ để tránh lặp nội dung, dễ truy vấn, dễ lọc dữ liệu thông qua khóa ngoại và bảng trung gian và hỗ trợ mở rộng trong tương lai.

### 3.4 Mô hình dữ liệu

#### 3.4.1 Danh sách các thực thể và mối kết hợp

Bảng 3.3 Danh sách các thực thể và mối kết hợp

STT	Tên thực thể/mối kết hợp	Diễn giải
1	users	Thực thể lưu tài khoản người dùng. Mỗi người dùng có một id duy nhất, username, mật khẩu, thời gian tạo.
2	notes	Thực thể lưu ghi chú do người dùng tạo. Mỗi note thuộc về một user, lưu title, content.
3	tags	Thực thể lưu các nhãn do người dùng tạo để phân loại ghi chú.

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

4	note_tags	Mối kết hợp giữa notes và tags có quan hệ N-N.
5	note_history	Thực thể lưu lịch sử chỉnh sửa ghi chú. Mỗi bản ghi lưu nội dung cũ của một note tại thời điểm edited_at.

### 3.4.2 Chi tiết các thực thể và mối kết hợp

Thực thể: users

Đặc tả: Bảng users dùng để lưu trữ thông tin của người dùng trong hệ thống. Mỗi người dùng khi đăng ký sẽ có một mã định danh duy nhất (id), cùng các thông tin như tên đăng nhập, mật khẩu, thời gian tạo tài khoản.

Bảng 3.4 Bảng users

Tên cột	Diễn giải	Kiểu dữ liệu	Khóa
id	Mã người dùng - mã định danh duy nhất cho mỗi user. Dùng trong FK ở các bảng khác(notes, tags), không thay đổi	INT AUTO_INCREMENT	PK
username	Tên đăng nhập (login)	VARCHAR(50)	
password	Lưu hash mật khẩu, đã mã hóa	VARCHAR(255)	
created_at	Ngày tạo tài khoản, lưu thời điểm tạo và cập nhật.	DATETIME	

Thực thể: notes

Đặc tả: Bảng notes lưu trữ thông tin các ghi chú cá nhân mà người dùng tạo. Mỗi ghi chú thuộc về một người dùng xác định và bao gồm tiêu đề, nội dung, xóa tạm thời, thời gian tạo và cập nhật. Đây là thực thể thể hiện nội dung chính của ứng dụng, nơi người dùng ghi lại và quản lý thông tin cá nhân của mình.

Bảng 3.5 Bảng notes

Tên cột	Diễn giải	Kiểu dữ liệu	Khóa
id	Mã ghi chú	INT AUTO_INCREMENT	PK
user_id	Xác định ghi chú thuộc người dùng nào	INT	FK => users.id

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

title	Tiêu đề ghi chú hiển thị trong danh sách.	VARCHAR(255)	
content	Nội dung ghi chú	TEXT	
is_deleted	Xóa tạm thời, giữ bản ghi để có thể khôi phục hoặc lưu lịch sử	TINYINT(1) DEFAULT 0	
created_at	Ngày tạo	DATETIME	
updated_at	Ngày cập nhật gần nhất	DATETIME	

### Thực thể: tags

Đặc tả: Bảng tags lưu thông tin các nhãn mà người dùng tạo để phân loại các ghi chú. Mỗi nhãn gắn liền với một người dùng và có tên nhãn duy nhất. Việc phân loại bằng nhãn giúp người dùng dễ dàng tìm kiếm và sắp xếp ghi chú theo chủ đề. Thực thể tags có quan hệ N-N với thực thể notes thông qua note\_tags.

Bảng 3.6 Bảng tags

Tên cột	Diễn giải	Kiểu dữ liệu	Khóa
id	Mã tag	INT AUTO_INCREMENT	PK
user_id	Xác định tag thuộc về người dùng nào	INT	FK=>user_id
tag_name	Tên tag	VARCHAR(50)	

### Thực thể: note\_tags

Đặc tả: Bảng note\_tags là bảng trung gian thể hiện mối quan hệ N-N giữa bảng notes và tags. Nhờ có thực thể này, hệ thống có thể quản lý linh hoạt việc gắn nhiều nhãn cho một ghi chú, hoặc sử dụng cùng một nhãn cho nhiều ghi chú khác nhau mà không làm trùng lặp dữ liệu.

Bảng 3.7 Bảng note\_tags

Tên cột	Diễn giải	Kiểu dữ liệu	Khóa
note_id	Ghi chú nào	INT	FK=>notes.id
tag_id	Tag nào	INT	FK=>tags.id

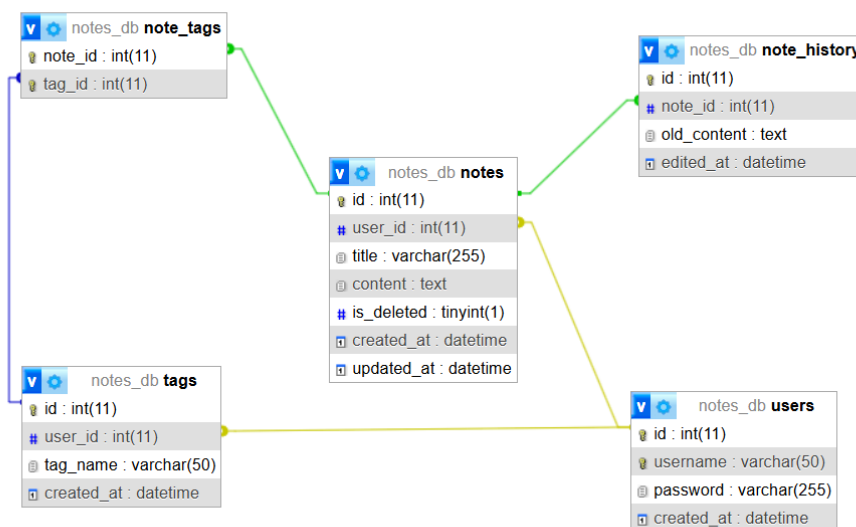
### Thực thể: note\_history

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động

Đặc tả: Bảng `note_history` dùng để lưu lại lịch sử chỉnh sửa của các ghi chú. Mỗi khi người dùng cập nhật nội dung của một ghi chú nào đó, hệ thống sẽ tạo một bản ghi mới trong bảng này, chứa nội dung cũ cùng với thời gian chỉnh sửa. Mục đích là giúp người dùng xem lại hoặc khôi phục phiên bản trước đó của ghi chú, theo dõi lịch sử thay đổi trong quá trình sử dụng.

Bảng 3.8 Bảng `note_history`

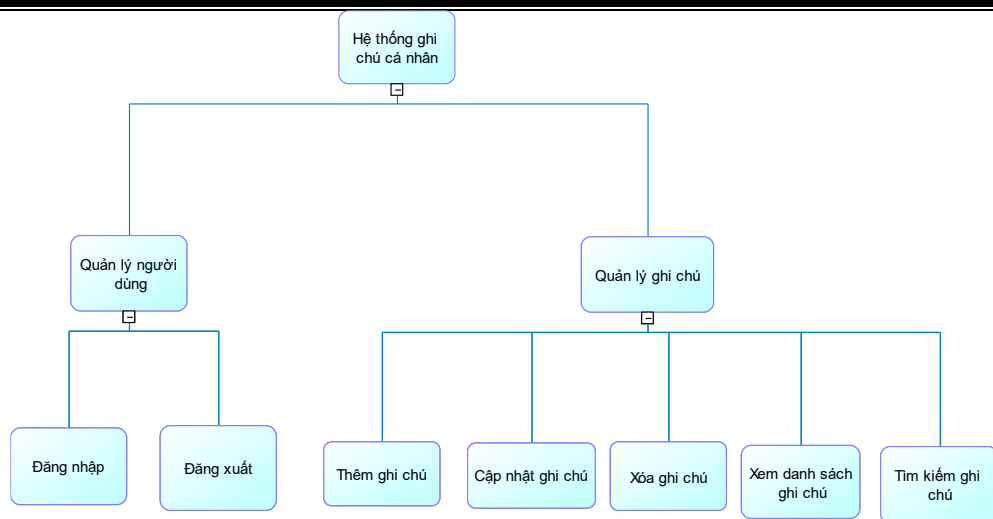
Tên cột	Diễn giải	Kiểu dữ liệu	Khóa
id	Mã lịch sử	INT AUTO_INCREMENT	PK
note_id	Ghi chú được chỉnh sửa	INT	FK=>notes.id
old_content	Nội dung cũ trước khi cập nhật	TEXT	
edited_at	Thời gian chỉnh sửa	DATETIME	



Hình 3.1 Mô hình ERD

### 3.5 Thiết kế xử lý

#### 3.5.1 Sơ đồ phân rã chức năng



Hình 3.2 Sơ đồ phân rã chức năng

Sơ đồ phân rã chức năng được thực hiện nhằm xác định các chức năng của hệ thống ghi chú cá nhân. Từ chức năng tổng quát, hệ thống được phân rã thành các chức năng chính bao gồm quản lý người dùng và quản lý ghi chú. Mỗi chức năng tiếp tục được phân rã thành các chức năng con như thêm, cập nhật, xóa, xem danh sách và tìm kiếm ghi chú. Việc phân rã chức năng giúp làm rõ phạm vi xử lý của hệ thống và là cơ sở để xây dựng mô hình DFD.

### 3.5.2 Mô hình DFD

Mô hình DFD được xây dựng nhằm mô tả luồng dữ liệu và các chức năng chính của hệ thống ghi chú cá nhân. Hệ thống được phân tích qua ba mức: mức ngữ cảnh, mức 1 và mức 2 nhằm làm rõ mối quan hệ giữa người dùng, các tiến trình xử lý và kho dữ liệu

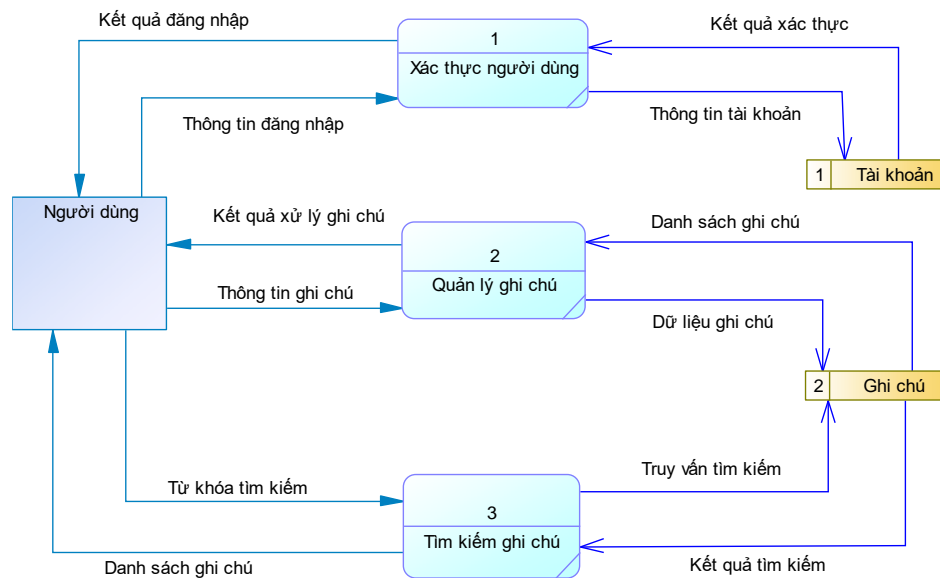
#### 3.5.2.1 Mô hình DFD mức ngữ cảnh



Hình 3.3 Mô hình DFD mức ngữ cảnh

Sơ đồ DFD mức ngữ cảnh được xây dựng nhằm mô tả tổng quan hệ thống ghi chú cá nhân, thể hiện mối quan hệ tương tác giữa hệ thống và người dùng. Trong sơ đồ này, người dùng là tác nhân chính, thực hiện các yêu cầu như đăng nhập, tạo ghi chú, chỉnh sửa, xóa và xem ghi chú. Hệ thống tiếp nhận các yêu cầu từ người dùng, tiến hành xử lý và trả về các kết quả tương ứng như thông tin xác thực, danh sách ghi chú hoặc thông báo trạng thái.

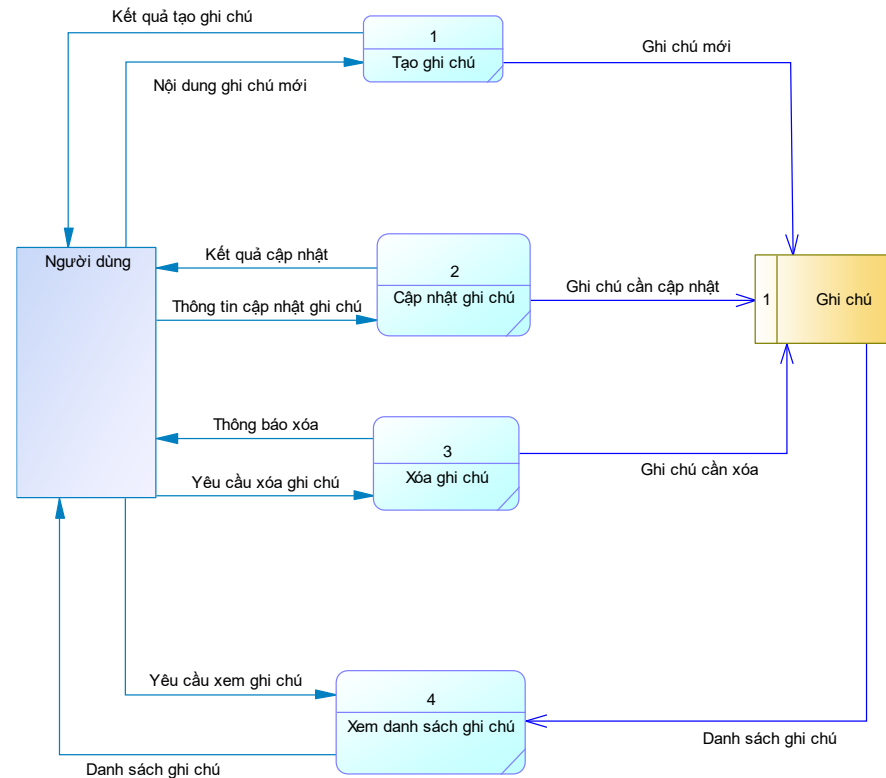
### 3.5.2.2 Mô hình DFD mức 1



Hình 3.4 Mô hình DFD mức 1

Sơ đồ DFD mức 1 được xây dựng nhằm phân rã tiến trình tổng quát của hệ thống ở mức ngữ cảnh thành các tiến trình xử lý chính, giúp làm rõ chức năng và luồng dữ liệu bên trong hệ thống. Cụ thể, hệ thống được chia thành các tiến trình như xác thực người dùng, quản lý ghi chú và tìm kiếm ghi chú. Các tiến trình này tương tác với kho dữ liệu tài khoản và kho dữ liệu ghi chú để thực hiện việc lưu trữ, truy xuất và xử lý thông tin.

### 3.5.2.3 Mô hình DFD mức 2



Hình 3.5 Mô hình DFD mức 2

Sơ đồ DFD mức 2 được xây dựng nhằm phân rã chi tiết tiến trình Quản lý ghi chú, bao gồm các chức năng: tạo, cập nhật, xóa và xem danh sách ghi chú. Qua đó mô tả rõ luồng dữ liệu giữa người dùng, các tiến trình xử lý và kho dữ liệu ghi chú của hệ thống.

### 3.5.3 Thiết kế xử lý hệ thống

Thiết kế xử lý mô tả cách hệ thống tiếp nhận, xử lý, phản hồi yêu cầu từ người dùng theo từng thao tác cụ thể. Hệ thống hoạt động dựa trên mô hình Client-Server, trong đó client đảm nhiệm nhận thao tác từ người dùng và server xử lý logic rồi trả kết quả lại. Hệ thống xử lý các chức năng theo quy trình:

#### 3.5.3.1 Đăng ký và đăng nhập

Khi người dùng đăng ký tài khoản, dữ liệu từ form sẽ được client gửi đi bằng phương thức POST. Server tiếp nhận yêu cầu, kiểm tra dữ liệu rỗng hay không hợp lệ, sau đó mã hóa mật khẩu rồi thực hiện truy vấn INSERT để lưu vào bảng users. Quá trình đăng nhập diễn ra tương tự, server nhận username và password từ POST, truy vấn MySQL để tìm username tương ứng, sau đó so sánh mật khẩu đã mã hóa

Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động trong database với mật khẩu người dùng nhập vào xem có đúng không. Nếu trùng khớp, server khởi tạo một session để xác định phiên làm việc của người dùng và cho phép truy cập các tính năng riêng tư của hệ thống.

#### 3.5.3.2 Xử lý ghi chú

Khi người dùng tạo mới ghi chú, server nhận tiêu đề (title) và nội dung (content), truy vấn INSERT để lưu vào notes, đồng thời gán user\_id tương ứng để xác định quyền sở hữu. Khi chỉnh sửa, trước khi cập nhật server sẽ lưu dữ liệu cũ vào note\_history cùng với thời gian chỉnh sửa edited\_at, sau đó cập nhật nội dung mới cho notes.id tương ứng. Để xóa tạm thời, không dùng DELETE mà thay vào đó dùng UPDATE is\_deleted = 1, vừa đảm bảo dữ liệu không mất vĩnh viễn, vừa giữ nguyên bản ghi phục vụ cho các chức năng lọc hoặc khôi phục sau này.

#### 3.5.3.3 Xử lý nhãn

Khi người dùng tạo một nhãn mới hoặc đổi tên nhãn, server nhận yêu cầu và dùng INSERT/UPDATE để thao tác trực tiếp với bảng tags. Việc gắn nhãn vào ghi chú cũng được xử lý bằng server, server nhận id của ghi chú và id của tag rồi INSERT vào bảng trung gian note\_tags để liên kết hai thực thể.

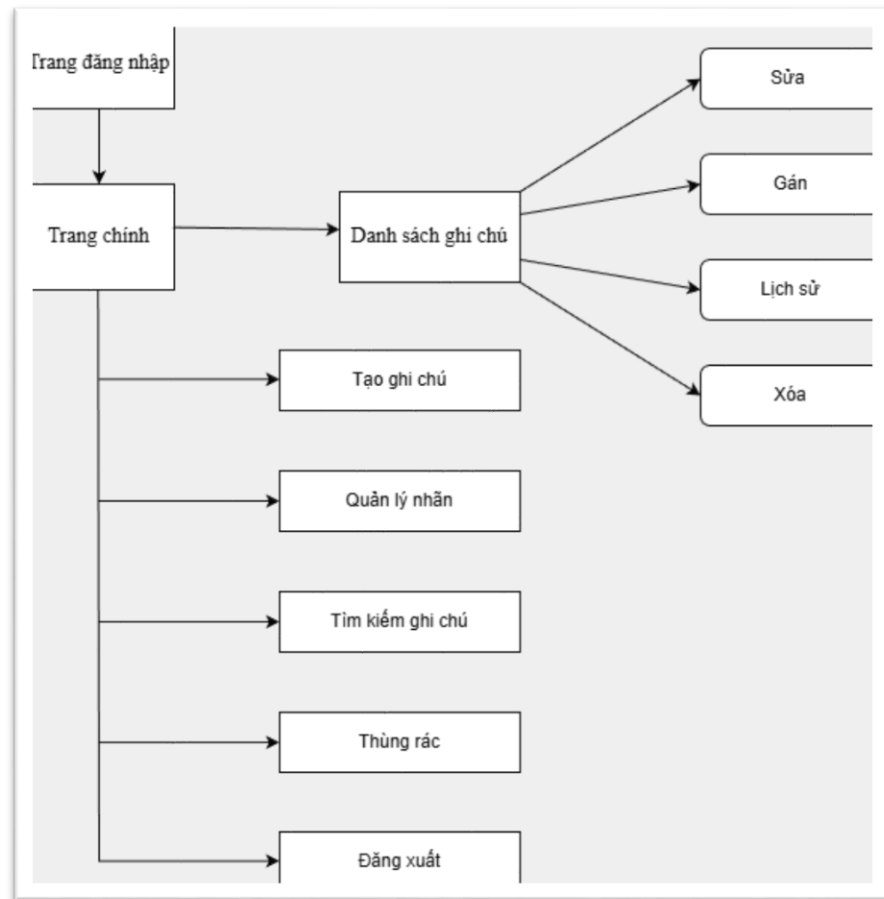
#### 3.5.3.4 Tìm kiếm ghi chú

Tính năng tìm kiếm ghi chú là thao tác xử lý trên server mà không thay đổi dữ liệu. Khi người dùng nhập từ khóa trong ô tìm kiếm, trình duyệt gửi yêu cầu đến server, server dùng SELECT kết hợp với điều kiện LIKE để tìm trong tiêu đề hoặc nội dung của những ghi chú thuộc quyền user đang đăng nhập và chưa bị xóa. Kết quả tìm thấy được sắp xếp theo thời gian cập nhật gần nhất trước khi trả về client để hiển thị.

### 3.6 Thiết kế giao diện

Thiết kế giao diện tập trung mô tả trải nghiệm trực quan và cách hệ thống hiển thị nội dung trên trình duyệt để giúp người dùng thao tác dễ dàng. Mục tiêu của giao diện trong hệ thống ghi chú cá nhân trực tuyến là đảm bảo đơn giản nhưng logic, thân thiện với người dùng.





Hình 3.6 Sitemap hệ thống ghi chú cá nhân

### 3.6.1 Giao diện đăng ký và đăng nhập

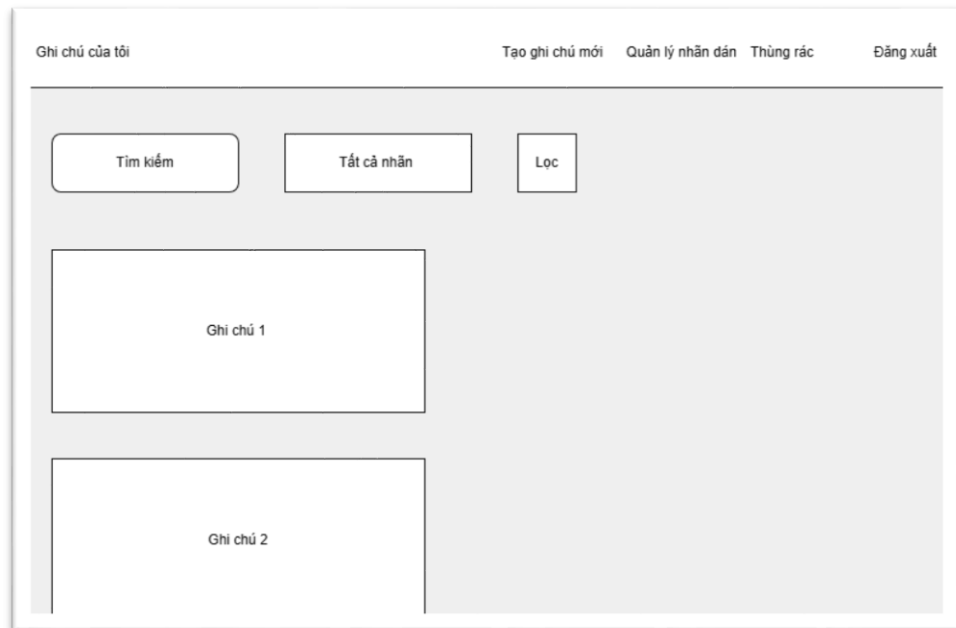
The mockup shows a simple login form with two input fields. The first field is labeled 'Tên đăng nhập:' (Username) and the second field is labeled 'Mật khẩu:' (Password). Both fields are empty rectangular boxes.

Hình 3.7 Phác thảo giao diện đăng ký và đăng nhập

Giao diện form đăng ký và đăng nhập được thiết kế tối giản với hai trường nhập thông tin cơ bản là tên đăng nhập và mật khẩu. Mỗi trường nhập được đặt trong khung có độ rộng vừa phải và căn giữa màn hình, giúp người dùng tập trung vào thao tác chính, đồng thời có vị trí hiển thị thông tin phản hồi như thông báo đăng ký thành

Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động công hoặc cảnh báo đăng nhập sai, giúp hệ thống tương tác với người dùng một cách rõ ràng và thân thiện.

### 3.6.2 Giao diện danh sách ghi chú



Hình 3.8 Phác thảo giao diện danh sách ghi chú

Danh sách ghi chú được hiển thị dưới dạng các khung thẻ (card) hoặc bảng danh sách đơn giản gồm tiêu đề và nội dung rút gọn, giúp người dùng dễ hình dung nội dung chính mà không bị quá tải thông tin. Các nút chức năng như sửa ghi chú, xem lịch sử, gán nhãn hoặc xóa tạm thời đều được đặt gần từng bản ghi, tạo cảm giác điều khiển trực tiếp đối tượng thay vì phải tìm kiếm nút thao tác ở nơi khác. Navbar của website được thiết kế đơn giản phù hợp với vị trí lọc theo nhãn hoặc hiển thị danh sách ghi chú đã xóa để người dùng dễ dàng phân loại nội dung theo nhu cầu của mình.

### 3.6.3 Giao diện tạo và chỉnh sửa ghi chú



Ghi chú của tôi

Tạo ghi chú mới

Tiêu đề:

Nội dung:

Hình 3.9 Phác thảo giao diện tạo và chỉnh sửa

Trang tạo hoặc chỉnh sửa ghi chú được bố trí với một ô nhập tiêu đề và một vùng textarea để nhập nội dung, các nút lưu hoặc quay lại được đặt ngay dưới vùng nhập giúp người dùng kiểm soát hành động rõ ràng.

### 3.6.4 Giao diện xem lịch sử



Ghi chú của tôi

Quay lại

Ghi chú 1

Khôi phục

Ghi chú 2

Khôi phục

Hình 3.10 Phác thảo giao diện xem lịch sử

Phần xem lịch sử chỉnh sửa được thiết kế đơn giản, các bản chỉnh sửa cũ được hiển thị theo trình tự thời gian rõ ràng, giúp hệ thống ghi chú không chỉ là nơi lưu nội dung mà còn là nơi có thể khôi phục lại lịch sử chỉnh sửa trước đó.

### 3.6.5 Giao diện quản lý nhãn

Id nhãn	Tên nhãn	Xóa
1	Nhãn 1	Xóa
2	Nhãn 2	Xóa
3	Nhãn 3	Xóa

Hình 3.11 Phác thảo giao diện quản lý nhãn

Giao diện quản lý nhãn cho phép người dùng tạo và quản lý các nhãn dùng để phân loại ghi chú. Việc sử dụng nhãn giúp hệ thống ghi chú được tổ chức khoa học hơn và hỗ trợ tìm kiếm nhanh chóng.

### 3.6.6 Giao diện gán nhãn

Chọn	Nhãn
<input type="radio"/>	Nhãn 1
<input type="radio"/>	Nhãn 2
<input type="radio"/>	Nhãn 3

Hình 3.12 Phác thảo giao diện gán nhãn

Giao diện gán nhãn cho phép người dùng có thể gán nhãn cho từng ghi chú, sau đó bấm vào nút lưu để lưu nhãn được gán vào.

### 3.6.7 Giao diện thùng rác



Hình 3.13 Phác thảo giao diện thùng rác

Giao diện thùng rác cho phép người dùng sẽ được xem danh sách các ghi chú đã được xóa trước đó.

Toàn bộ bố cục giao diện được xây dựng với các thành phần tự động co giãn theo kích thước màn hình, khi trên máy tính có thể hiển thị nội dung thành 2 cột, khi trên điện thoại thì co lại 1 cột full-width để dễ đọc và dễ thao tác cảm ứng, mang lại trải nghiệm tốt mà không cần mô tả dài dòng phức tạp.

## 3.7 Kết chương

Chương 3 đã trình bày toàn bộ quá trình thiết kế hệ thống website ghi chú cá nhân trực tuyến. Các thành phần dữ liệu được mô hình hóa tối giản với 5 bảng trong MySQL, đồng thời đảm bảo ràng buộc quan hệ chặt chẽ bằng khóa chính và khóa ngoại. Các chức năng xử lý quan trọng như đăng ký, đăng nhập, thao tác với ghi chú, gắn nhãn và lưu lịch sử chỉnh sửa được thiết kế rõ ràng theo nguyên tắc Client–Server và triển khai bằng PHP-MySQLi trong môi trường XAMPP.

Ngoài ra, giao diện hệ thống tập trung vào tính tối giản và trực quan, thân thiện với người dùng, sử dụng Bootstrap để chuẩn hóa bảng biểu, nút bấm, đảm bảo trải nghiệm mượt mà và thích hợp trên nhiều thiết bị. Những thiết kế này là cơ sở quan trọng để hệ thống có thể mở rộng trong tương lai.

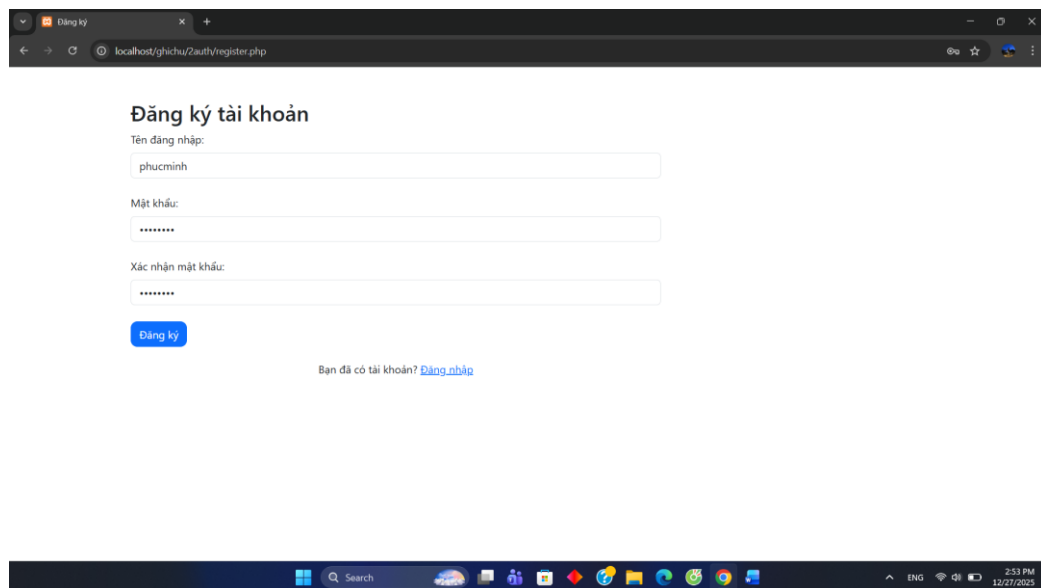
## CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

### 4.1 Kết quả xây dựng giao diện hệ thống

Chương 4 trình bày các kết quả đạt được sau quá trình phân tích, thiết kế và hiện thực hóa hệ thống ghi chú cá nhân trực tuyến. Nội dung tập trung mô tả giao diện hệ thống, các chức năng chính đã triển khai và đánh giá mức độ đáp ứng so với yêu cầu đặt ra ban đầu.

Giao diện hệ thống ghi chú cá nhân trực tuyến được xây dựng theo hướng đơn giản, dễ sử dụng và thân thiện với người dùng. Trong quá trình triển khai, hệ thống sử dụng HTML để xây dựng cấu trúc trang, CSS và Bootstrap để thiết kế giao diện, đảm bảo tính thống nhất và khả năng hiển thị tốt trên nhiều kích thước màn hình khác nhau. Các thành phần giao diện được bố trí rõ ràng, giúp người dùng dễ dàng thao tác ngay từ lần sử dụng đầu tiên.

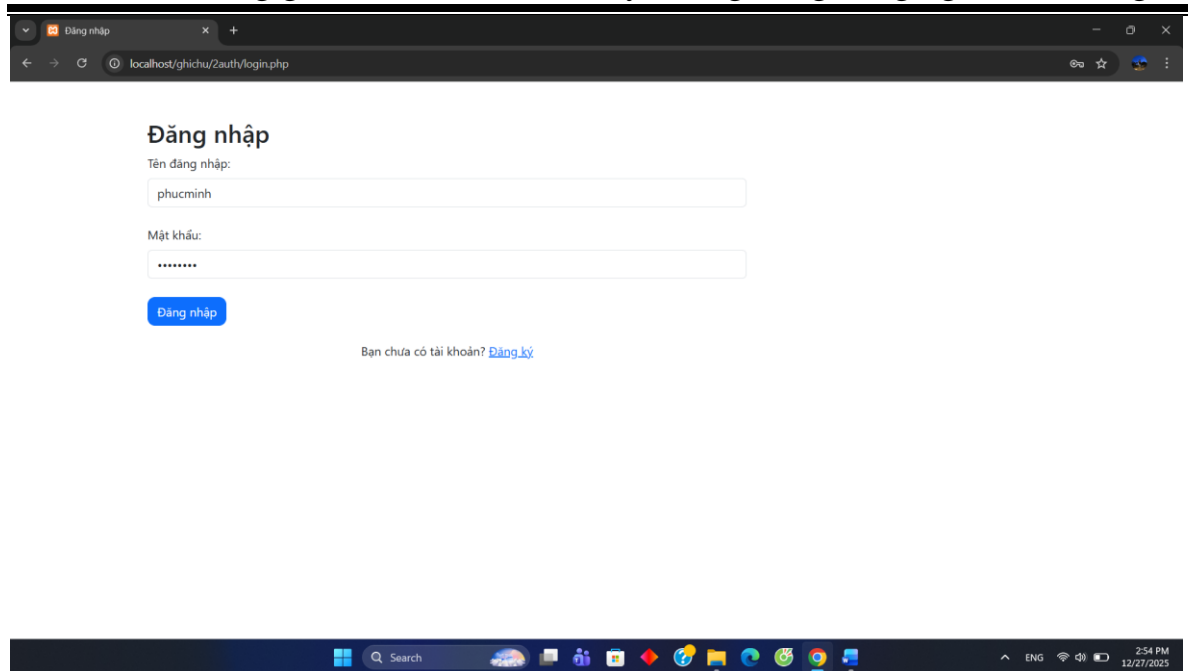
#### 4.1.1 Giao diện đăng ký và đăng nhập



Hình 4.1 Giao diện đăng ký

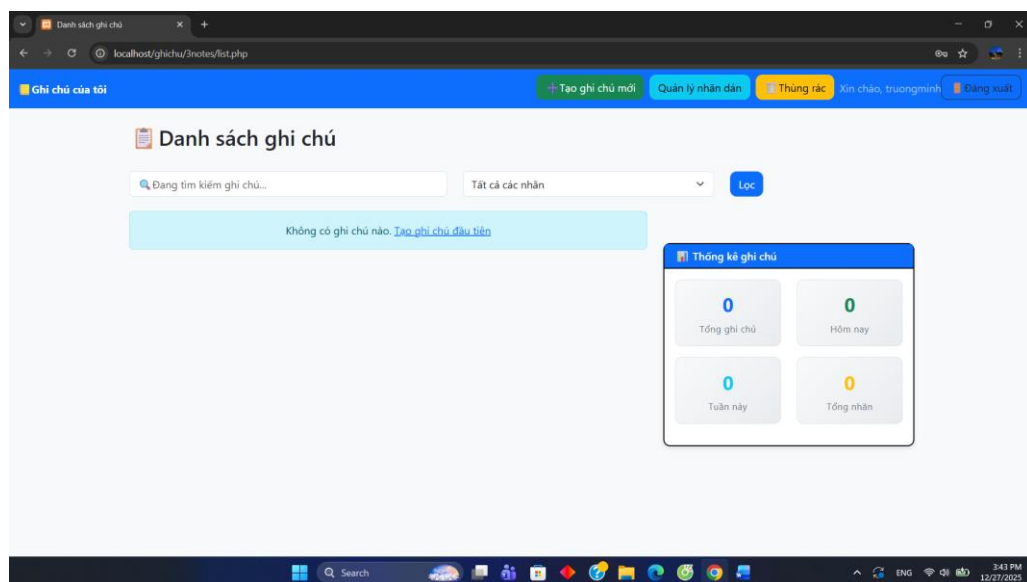
Giao diện đăng ký hỗ trợ người dùng tạo tài khoản mới với các trường thông tin cơ bản, dữ liệu được kiểm tra trước khi lưu vào cơ sở dữ liệu nhằm đảm bảo tính chính xác. Người dùng chỉ cần nhập tài khoản, mật khẩu, xác nhận mật khẩu và nhấp vào nút đăng ký là có thể qua phần đăng nhập vào trang web.

## Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động



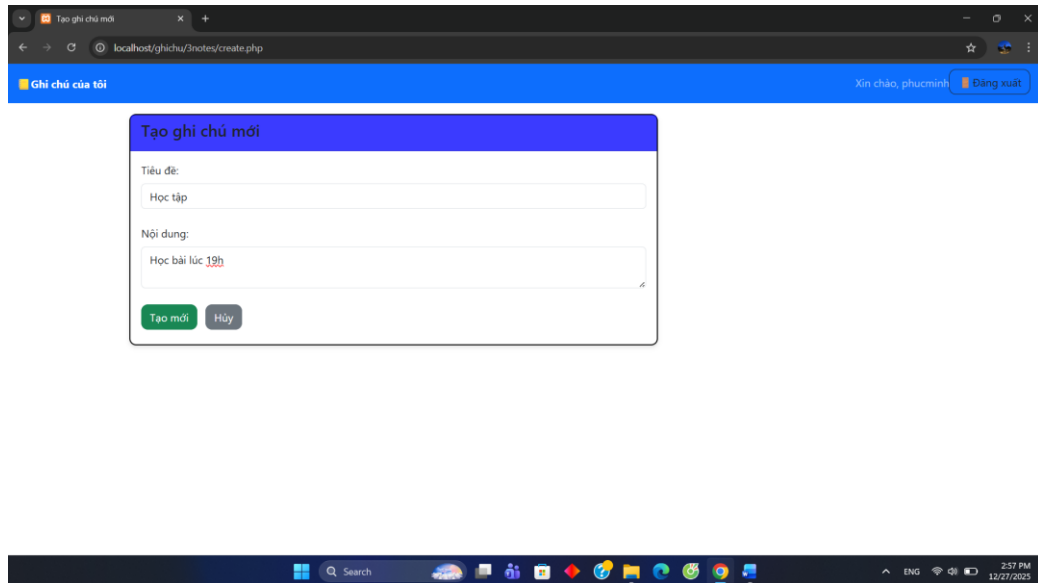
Hình 4.2 Giao diện đăng nhập

Giao diện đăng nhập cho phép người dùng nhập thông tin tài khoản gồm tên đăng nhập và mật khẩu để truy cập vào hệ thống. Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập vào và hiển thị thông báo khi đăng nhập thành công hoặc thất bại. Nếu thành công thì sẽ được chuyển đến phần giao diện chính như hình 4.3.



Hình 4.3 Giao diện chính

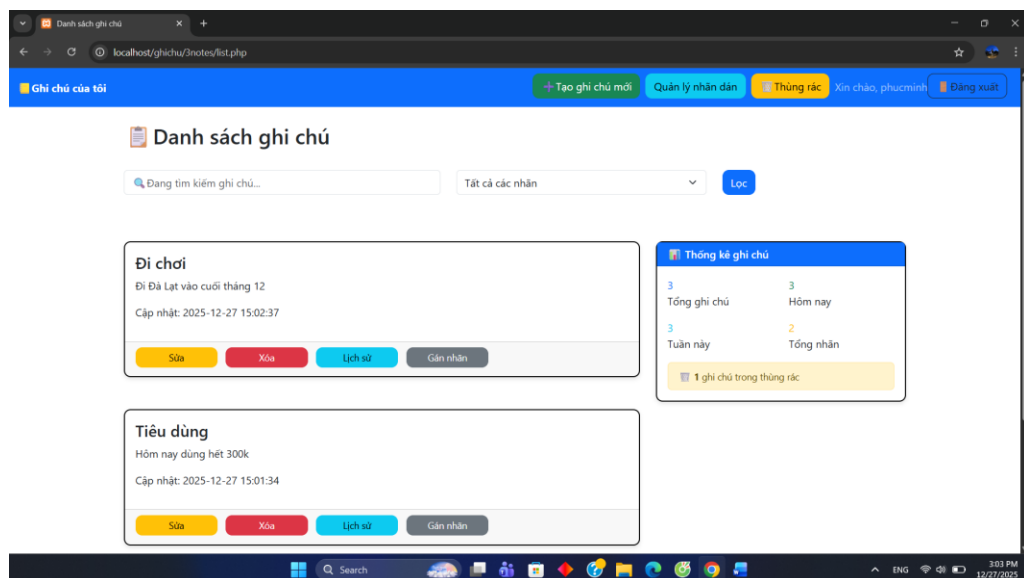
## 4.1.2 Giao diện tạo ghi chú



Hình 4.4 Giao diện tạo ghi chú

Giao diện tạo ghi chú được thiết kế dưới dạng biểu mẫu (form) gồm các trường tiêu đề và nội dung. Người dùng có thể nhập thông tin ghi chú một cách thuận tiện. Sau khi gửi biểu mẫu, hệ thống sẽ xử lý dữ liệu và lưu trữ vào cơ sở dữ liệu MySQL, đồng thời hiển thị thông báo để người dùng biết trạng thái thực hiện.

## 4.1.3 Giao diện danh sách ghi chú



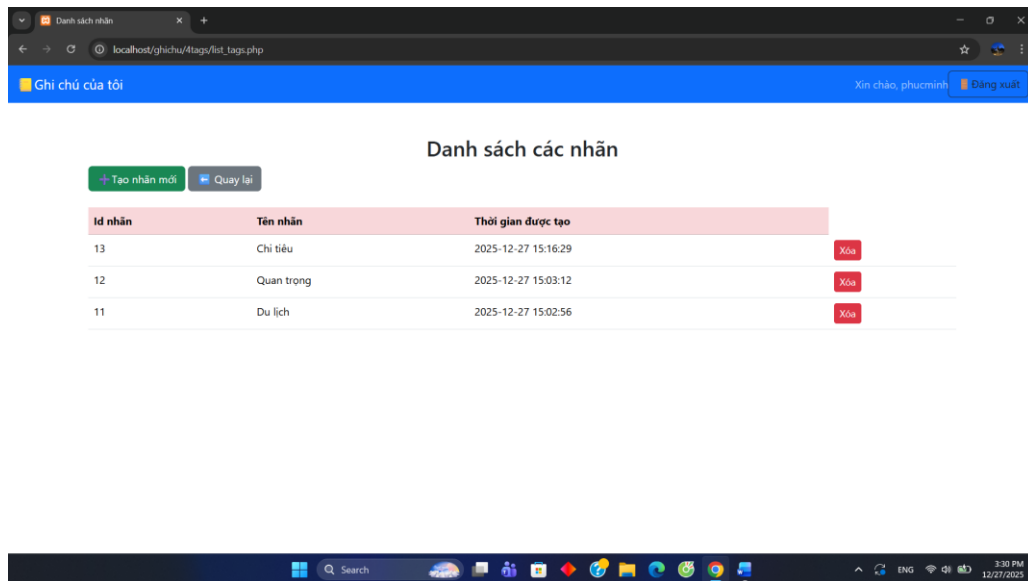
Hình 4.5 Giao diện danh sách ghi chú

Sau khi tạo ghi chú thành công, người dùng được chuyển đến giao diện danh sách ghi chú. Tại đây, hệ thống hiển thị toàn bộ các ghi chú thuộc về tài khoản đang được đăng nhập. Các ghi chú được sắp xếp theo thời gian tạo hoặc cập nhật gần nhất,



Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động giúp người dùng dễ dàng theo dõi và quản lý nội dung. Mỗi ghi chú đều có các nút chức năng như xem lịch sử, gắn nhãn, chỉnh sửa và xóa.

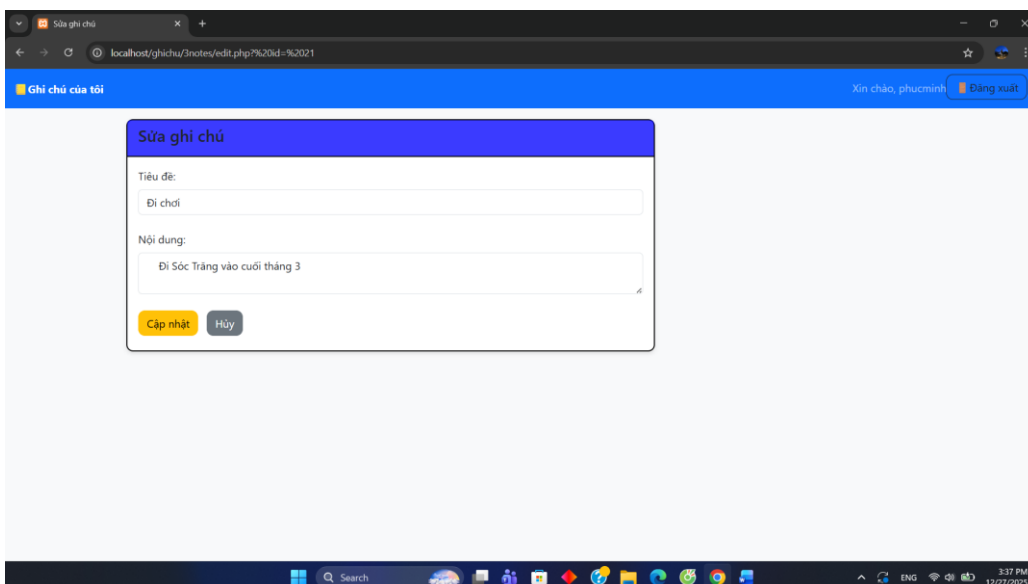
#### 4.1.4 Giao diện quản lý nhãn ghi chú



Hình 4.6 Giao diện quản lý nhãn ghi chú

Trong giao diện quản lý nhãn, người dùng có thể tạo nhãn và xem danh sách các nhãn ghi chú mà người dùng đã tạo gồm có id nhãn, tên nhãn và thời gian được tạo, sau đó có thể xóa các nhãn tùy ý người dùng.

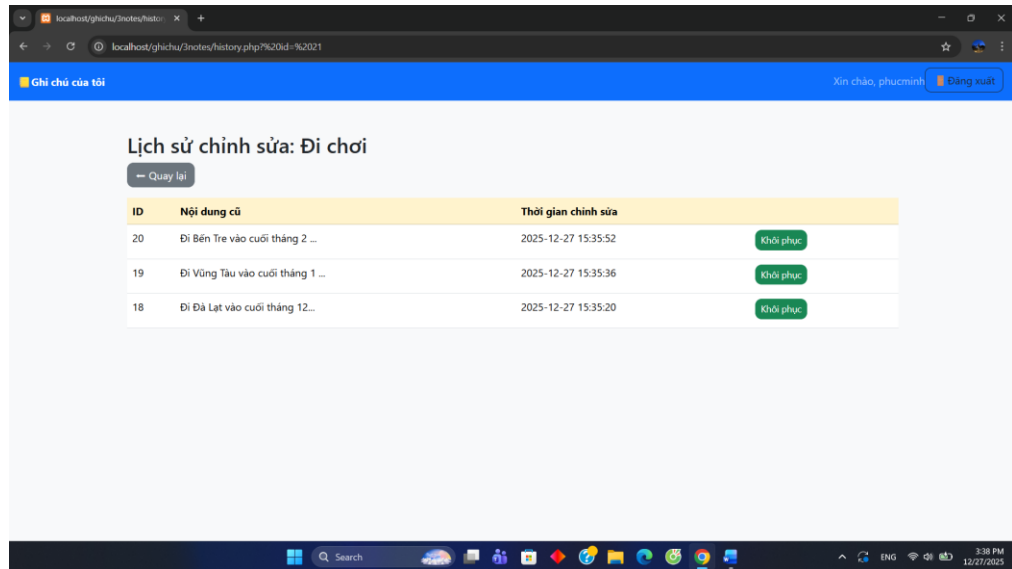
#### 4.1.5 Giao diện sửa ghi chú



Hình 4.7 Giao diện sửa ghi chú

Tại đây, người dùng có thể chỉnh sửa ghi chú theo ý muốn, gồm có tiêu đề và nội dung. Sau đó, nhấn vào nút cập nhật để thay đổi thành ghi chú mới nhất.

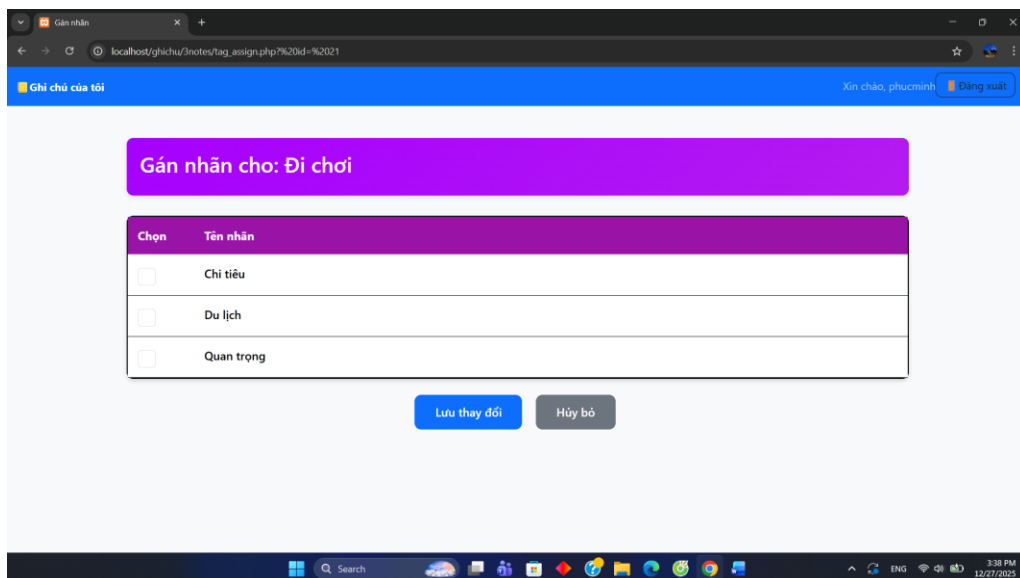
#### 4.1.6 Giao diện xem lịch sử ghi chú



Hình 4.8 Giao diện xem lịch sử

Trong giao diện này, người dùng có thể xem lịch sử các ghi chú trước đó gồm có các cột id, nội dung cũ và thời gian chỉnh sửa. Nếu muốn khôi phục lại thành ghi chú cũ thì chỉ cần nhấp vào nút “Khôi phục”.

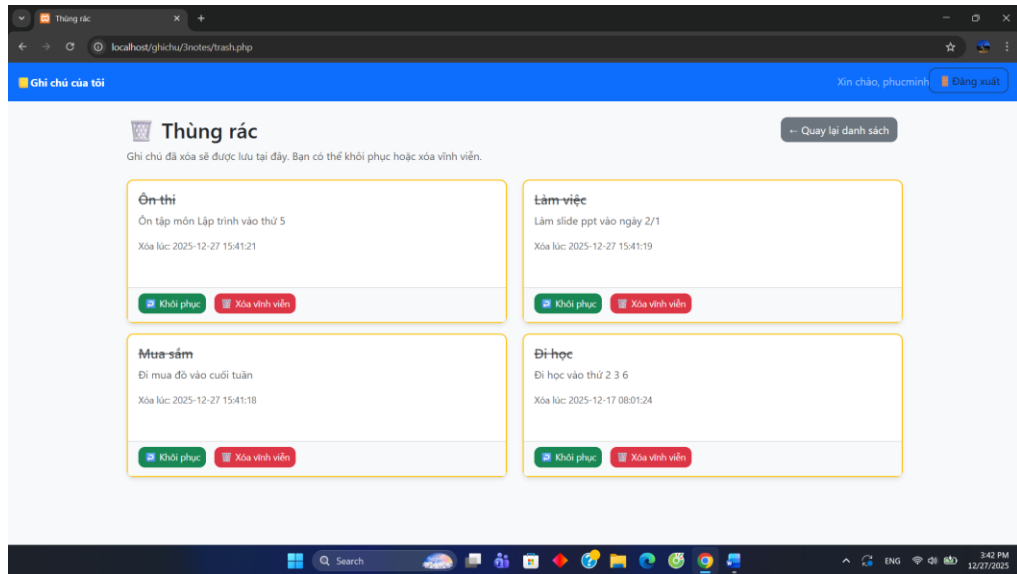
#### 4.1.7 Giao diện gán nhãn ghi chú



Hình 4.9 Giao diện gán nhãn

Trong giao diện gán nhãn, người dùng có thể gán nhãn cho ghi chú của mình bằng cách đánh dấu vào từng nhãn do người dùng đã tạo ra.

#### 4.1.8 Giao diện thùng rác chứa ghi chú



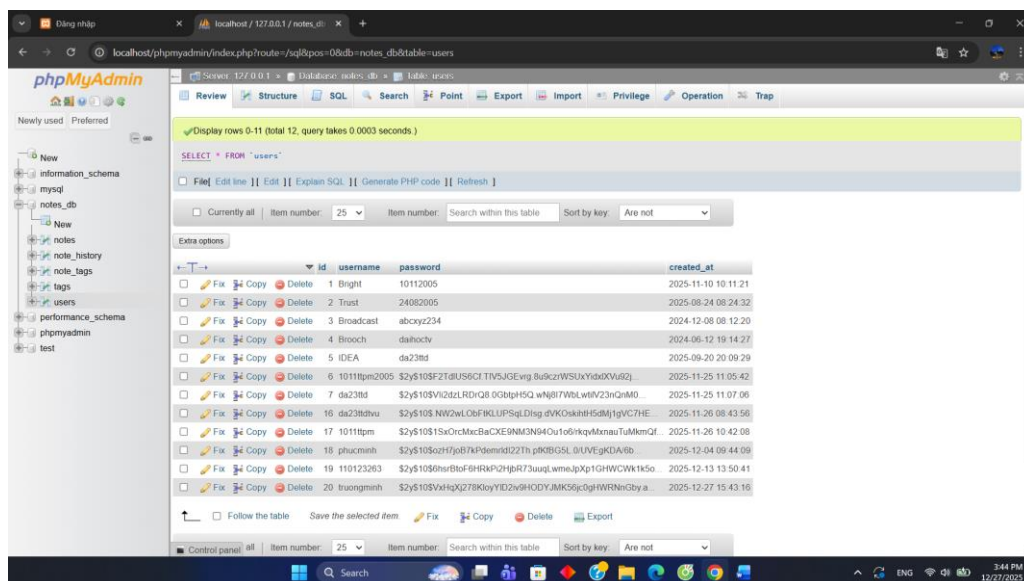
Hình 4.10 Giao diện thùng rác

Tại đây, người dùng sẽ được xem danh sách các ghi chú đã được xóa trước đó vào thời gian nào, người dùng có thể khôi phục lại các ghi chú nếu muốn bằng cách nhấp vào nút “Khôi phục” hoặc “Xóa vĩnh viễn” để xóa vĩnh viễn các ghi chú.

#### 4.2 Kết quả xây dựng các chức năng chính

Dựa trên các yêu cầu chức năng đã được xác định ở Chương 3, hệ thống ghi chú cá nhân trực tuyến đã được xây dựng với các chức năng cốt lõi, đáp ứng nhu cầu lưu trữ và quản lý ghi chú của người dùng. Các chức năng được triển khai theo mô hình Client–Server, trong đó PHP xử lý logic phía máy chủ và MySQL đảm nhiệm vai trò lưu trữ dữ liệu.

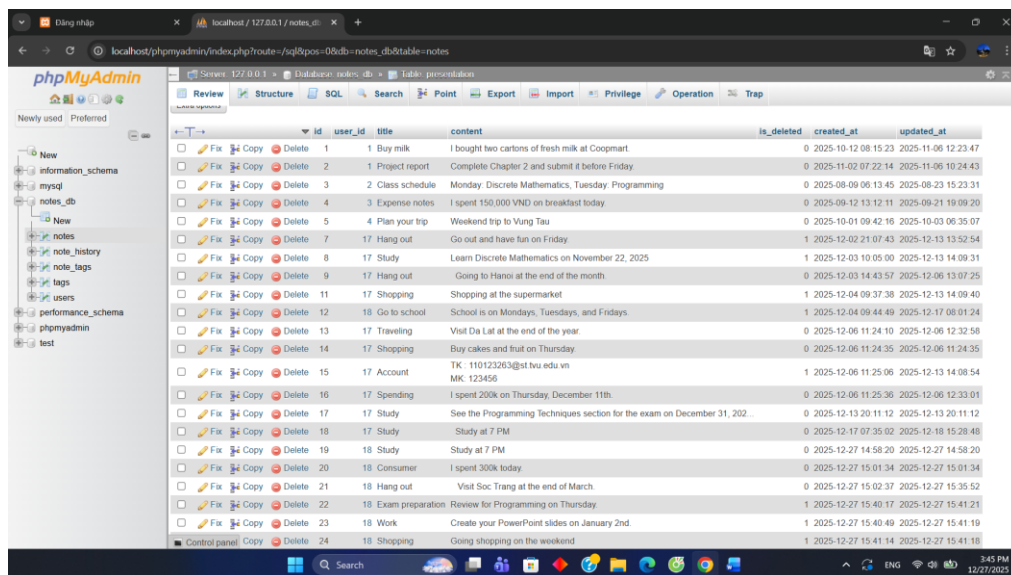
## 4.2.1 Chức năng quản lý tài khoản người dùng



Hình 4.11 Chức năng quản lý tài khoản

Hệ thống cho phép người dùng đăng ký tài khoản, đăng nhập và đăng xuất khỏi hệ thống. Thông tin tài khoản được lưu trữ trong cơ sở dữ liệu và được kiểm tra trong quá trình xác thực người dùng. Việc quản lý phiên đăng nhập giúp đảm bảo rằng mỗi người dùng chỉ có thể truy cập và thao tác trên các ghi chú của riêng mình.

## 4.2.2 Chức năng quản lý ghi chú

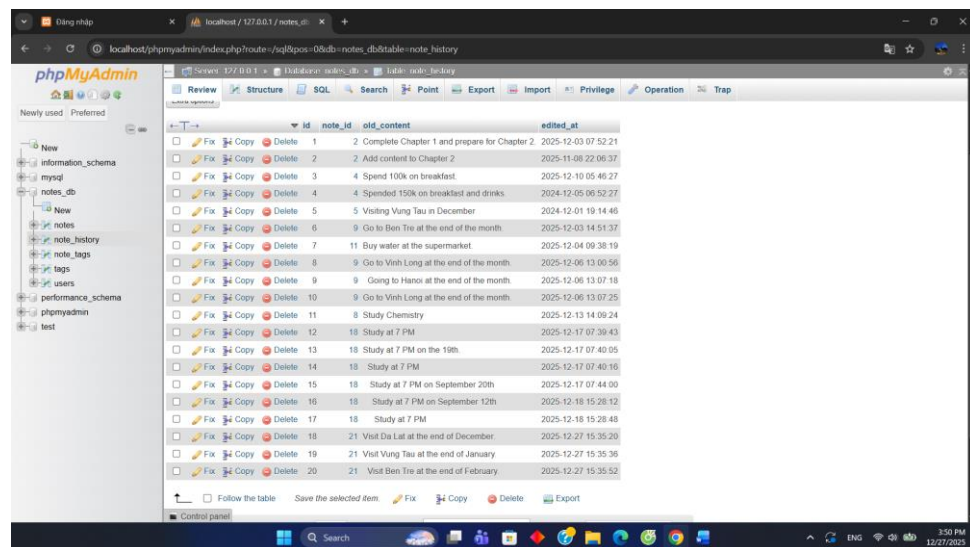


Hình 4.12 Chức năng quản lý ghi chú

Chức năng quản lý ghi chú là chức năng chính của hệ thống. Người dùng có thể tạo mới ghi chú, chỉnh sửa nội dung ghi chú đã có và xóa ghi chú khi không còn nhu cầu sử dụng. Thao tác xóa được thực hiện theo hình thức xóa tạm thời, nghĩa là

Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động dữ liệu không bị xóa hoàn toàn khỏi cơ sở dữ liệu mà chỉ được đánh dấu, giúp tránh mất dữ liệu ngoài ý muốn và có thể khôi phục khi cần thiết.

#### 4.2.3 Chức năng lưu lịch sử



Hình 4.13 Chức năng lưu lịch sử ghi chú

Với chức năng lưu lịch sử ghi chú, người dùng có thể xem lại nội dung ghi chú cũ để có thể phục hồi ghi chú hoặc xóa vĩnh viễn ghi chú.

#### 4.3 Đánh giá kết quả đạt được

Sau quá trình triển khai, đồ án đã xây dựng thành công hệ thống ghi chú cá nhân trực tuyến dựa trên công nghệ web động, sử dụng PHP để xử lý logic phía máy chủ và MySQL để lưu trữ dữ liệu. Hệ thống được triển khai và vận hành ổn định trong môi trường XAMPP, đáp ứng các yêu cầu cơ bản của một ứng dụng ghi chú cá nhân. Các chức năng chính như đăng nhập người dùng, tạo mới, chỉnh sửa, xóa ghi chú (xóa mềm) và hiển thị danh sách ghi chú theo từng tài khoản đã được hiện thực hóa đầy đủ, đúng với mục tiêu và yêu cầu đã đề ra trong Chương 3.

Bên cạnh những kết quả đạt được, hệ thống vẫn còn một số hạn chế như chưa hỗ trợ tìm kiếm nâng cao, phân loại ghi chú theo nhãn hoặc đồng bộ dữ liệu trên nhiều thiết bị. Tuy nhiên, với phạm vi của một đồ án cơ sở ngành, các kết quả đạt được là phù hợp và mang tính thực tiễn. Đồ án đã thể hiện khả năng vận dụng kiến thức về công nghệ web, lập trình PHP và quản lý cơ sở dữ liệu MySQL vào việc xây dựng một ứng dụng web hoàn chỉnh, đồng thời tạo nền tảng để tiếp tục mở rộng và phát triển hệ thống trong tương lai.

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

Trong bối cảnh công nghệ web ngày càng phát triển, việc xây dựng các hệ thống ứng dụng trực tuyến phục vụ nhu cầu cá nhân và học tập trở nên phổ biến. Đề tài “*Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động*” được thực hiện với mục tiêu giúp sinh viên tiếp cận quy trình xây dựng một ứng dụng web hoàn chỉnh, từ khâu phân tích yêu cầu đến thiết kế và triển khai hệ thống.

Thông qua quá trình nghiên cứu và thực hiện đồ án, hệ thống ghi chú cá nhân trực tuyến đã được xây dựng thành công và vận hành ổn định trong môi trường XAMPP. Hệ thống đáp ứng các chức năng cơ bản như đăng ký, đăng nhập người dùng; tạo mới, chỉnh sửa, xóa ghi chú; lưu trữ dữ liệu bằng MySQL và hiển thị giao diện thân thiện thông qua HTML, CSS, JavaScript và Bootstrap. Các chức năng được triển khai theo mô hình Client–Server, đảm bảo tính rõ ràng và phù hợp với kiến thức nền tảng của ngành Công nghệ thông tin.

Bên cạnh kết quả về mặt hệ thống, đồ án còn giúp em củng cố và nâng cao kiến thức về lập trình web động, quản lý cơ sở dữ liệu và tổ chức mã nguồn. Quá trình thực hiện đề tài góp phần rèn luyện tư duy phân tích, kỹ năng thiết kế hệ thống và khả năng áp dụng kiến thức lý thuyết vào thực tiễn, đáp ứng mục tiêu của một đồ án cơ sở ngành.

### 5.2 Hướng phát triển

Mặc dù hệ thống đã đáp ứng được các yêu cầu cơ bản của một website ghi chú cá nhân, tuy nhiên vẫn còn nhiều khả năng mở rộng và phát triển trong tương lai. Trước hết, hệ thống có thể được nâng cấp với các chức năng nâng cao như tìm kiếm ghi chú theo từ khóa, phân loại ghi chú theo thẻ (tags), sắp xếp ghi chú theo thời gian hoặc mức độ ưu tiên nhằm tăng hiệu quả quản lý dữ liệu cho người dùng.

Bên cạnh đó, hệ thống có thể được mở rộng về mặt công nghệ và bảo mật. Việc bổ sung các cơ chế xác thực nâng cao, mã hóa mật khẩu mạnh hơn, phân quyền người dùng hoặc tích hợp đăng nhập thông qua tài khoản mạng xã hội sẽ giúp nâng cao mức độ an toàn và tính tiện lợi. Ngoài ra, hệ thống có thể được triển khai trên máy chủ

Thiết kế hệ thống ghi chú cá nhân trực tuyến ứng dụng công nghệ web động thực tế hoặc tích hợp công nghệ lưu trữ đám mây để hỗ trợ truy cập và đồng bộ dữ liệu trên nhiều thiết bị.

Trong tương lai, đề tài có thể được phát triển thành một ứng dụng hoàn chỉnh hơn với giao diện hiện đại, hỗ trợ đa nền tảng và khả năng mở rộng linh hoạt. Đây sẽ là tiền đề quan trọng để tiếp tục nghiên cứu sâu hơn trong các đề án chuyên ngành hoặc ứng dụng thực tế trong học tập và công việc.

**TÀI LIỆU THAM KHẢO**

- [1] Fpt shop, “Google Keep.” 2025. [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/so-sanh-cac-ung-dung-ghi-chu-google-keep-evernote-va-onenote-149903>
- [2] Fpt shop, “Evernote.” 2025. [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/so-sanh-cac-ung-dung-ghi-chu-google-keep-evernote-va-onenote-149903>
- [3] Base.vn, “Notion.” 2025. [Online]. Available: <https://base.vn/blog/notion-la-gi/>
- [4] Nguyễn Văn Vy, “Giáo trình Lập trình Web PHP & MySQL,” 2020.
- [5] Đỗ Xuân Lôi, “Giáo trình Cơ sở dữ liệu,” 2019.
- [6] Nguyễn Thị Thanh Hà, “Nhập môn Công nghệ Web,” 2019.
- [7] Trần Ngọc Thái Sơn, “Lập trình Web Front-End với HTML5, CSS3, JavaScript,” 2022.
- [8] R. Nixon, *Learning PHP, MySQL & JavaScript*, 5th Edition. O'Reilly Media, 2018.
- [9] Jon Duckett, *JavaScript & jQuery*. 2014.
- [10] W3School, “W3Schools Online.” Nov. 15, 2025. [Online]. Available: <https://www.w3schools.com>
- [11] PHP Manual, “PHP Documentation.” 2024. [Online]. Available: [php.net/manual/en/](http://php.net/manual/en/)



**PHỤ LỤC (NẾU CÓ)**