

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

~~~~~\*~~~~~



**TRƯỜNG ĐẠI HỌC**  
**Sư phạm Thành phố Hồ Chí Minh**  
**HCMC University of Education**



# **BÁO CÁO ĐỒ ÁN**

**ĐỀ TÀI**  
**XÂY DỰNG TRANG WEB QUẢN LÝ CÔNG VIỆC**

**Học phần: COMP130301 – Phát triển ứng dụng Web**

**TP HỒ CHÍ MINH – 4/2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**

~~~~~\*~~~~~

BÁO CÁO ĐỒ ÁN

ĐỀ TÀI **XÂY DỰNG TRANG WEB QUẢN LÝ CÔNG VIỆC**

Học phần: COMP130301 – Phát triển ứng dụng Web

| | | | | |
|-----------------------|---|--------------------|---|---------------|
| Sinh viên thực hiện 1 | : | Hứa Tiên Hào | - | 46.01.104.049 |
| Sinh viên thực hiện 2 | : | Huỳnh Thị Yến Khoa | - | 46.01.104.087 |
| Sinh viên thực hiện 3 | : | Phước Công Nguyên | - | 46.01.104.125 |
| Sinh viên thực hiện 4 | : | Châu Tiểu My | - | 47.01.104.131 |
| Sinh viên thực hiện 5 | : | Hoàng Văn Thịnh | - | 47.01.104.201 |

Giảng viên hướng dẫn: ThS. Ma Ngân Giang

TP HỒ CHÍ MINH – 4/2023

(Của giảng viên hướng dẫn)

[illegible]

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời tri ân chân thành và sâu sắc nhất đến cô *Ma Ngân Giang*. Trong quá trình tìm hiểu và học tập bộ môn *Phát triển ứng dụng Web*, chúng em đã nhận được sự giảng dạy và hướng dẫn rất tận tình, tâm huyết của cô. Cô đã giúp chúng em tích lũy thêm nhiều kiến thức hay và bổ ích để phục vụ cho những học phần liên quan, chuyên môn nghề nghiệp cũng như trong cuộc sống. Từ những kiến thức mà cô truyền đạt, chúng em xin trình bày lại những gì mình đã tìm hiểu và được học từ cô để giải quyết đề tài “Xây dựng trang web quản lý công việc”.

Tuy nhiên, kiến thức về bộ môn *Phát triển ứng dụng Web* của chúng em vẫn còn những hạn chế nhất định. Do đó, không thể tránh khỏi những thiếu sót trong quá trình hoàn thành đồ án môn học này. Mong cô xem và góp ý để bài báo cáo học phần của nhóm chúng em được hoàn thiện hơn.

Kính chúc cô hạnh phúc và thành công hơn nữa trong sự nghiệp “trồng người” - dìu dắt những thế hệ học trò hoàn thành những ước mơ của mình. Kính chúc cô và gia đình nhiều sức khỏe.

Chúng em xin chân thành cảm ơn!

Sinh viên,

Hứa Tiên Hào

Huỳnh Thị Yến Khoa

Phước Công Nguyên

Châu Tiểu My

Hoàng Văn Thịnh

BẢNG PHÂN CÔNG NHIỆM VỤ

Tự đánh giá xếp hạng đóng góp của các thành viên trong nhóm.

| STT | Mã số sinh viên | Họ và tên | Công việc | Mức độ tham gia |
|-----|-----------------|--------------------|---|-----------------|
| 1 | 46.01.104.049 | Hứa Tiên Hào | Code chính
Thiết kế CSDL
Thực hiện thiết kế layout | 100% |
| 2 | 46.01.104.087 | Huỳnh Thị Yến Khoa | Tìm hiểu cơ sở lý thuyết
Lên ý tưởng layout
Code phụ phần task-board
Thuyết trình | 90% |
| 3 | 46.01.104.125 | Phước Công Nguyên | Tìm hiểu và thu thập cơ sở lý thuyết
Lên ý tưởng layout
Code phụ phần task-board
Viết báo cáo bản Word | 90% |
| 4 | 47.01.104.131 | Châu Tiểu My | Code phụ: Todo-list | 70% |
| 5 | 47.01.104.201 | Hoàng Văn Thịnh | Code phụ: Todo-list | 0% |

MỤC LỤC

| | |
|--|----|
| CHƯƠNG 1. GIỚI THIỆU | 1 |
| 1.1. Lý do chọn đề tài | 1 |
| 1.2. Mục tiêu của đề tài | 1 |
| 1.3. Phạm vi đề tài | 1 |
| 1.4. Phương thức thực hiện..... | 1 |
| CHƯƠNG 2. CƠ SỞ LÝ THUYẾT..... | 3 |
| 2.1. Lý thuyết cơ sở dữ liệu | 3 |
| 2.1.1. Khái niệm | 3 |
| 2.1.2. Ưu điểm | 4 |
| 2.1.3. Nhược điểm | 4 |
| 2.1.4. Phân loại cơ sở dữ liệu | 4 |
| 2.2. Hệ quản trị cơ sở dữ liệu MySQL | 5 |
| 2.3. Ngôn ngữ lập trình xây dựng hệ thống..... | 5 |
| 2.3.1. HTML và CSS | 5 |
| 2.3.2. Ngôn ngữ lập trình JavaScript và PHP..... | 7 |
| 2.4. XAMPP | 9 |
| CHƯƠNG 3. CHỨC NĂNG CỦA HỆ THỐNG | 11 |
| 3.1. Chức năng đăng nhập | 11 |
| 3.2. Chức năng đăng kí..... | 12 |
| 3.3. Chức năng đăng xuất | 16 |
| 3.4. Todo-list..... | 17 |
| 3.5. Task-board | 18 |
| CHƯƠNG 4. GIAO DIỆN CỦA HỆ THỐNG | 20 |
| 4.1. Trang chủ..... | 20 |

| | |
|--|----|
| 4.2. Đăng ký..... | 26 |
| 4.3. Đăng nhập..... | 28 |
| 4.4. Todo-lists | 29 |
| 4.5. Task-board | 30 |
| 4.6. Đăng xuất..... | 32 |
| CHƯƠNG 5. KẾT LUẬN | 33 |
| 5.1. Những công việc đã làm được..... | 33 |
| 5.2. Hướng phát triển của đồ án quản lý công việc trên web | 33 |

DANH MỤC HÌNH ẢNH

| | |
|---|----|
| Hình 2.1. Cơ sở dữ liệu (database) | 3 |
| Hình 2.2. MySQL là gì? | 5 |
| Hình 2.3. Bố cục cơ bản của một trang HTML | 6 |
| Hình 2.4. Ngôn ngữ CSS | 7 |
| Hình 2.5. JavaScript là gì? | 7 |
| Hình 2.6. Ngôn ngữ PHP | 8 |
| Hình 2.7. Giao diện chính của Xampp | 9 |
| Hình 3.1. Đoạn code tạo form đăng nhập “form-box login” | 11 |
| Hình 3.2. Đoạn code function loginUser | 11 |
| Hình 3.3. Đoạn code file login.inc.php | 12 |
| Hình 3.4. Cấu trúc SQL của bảng users | 12 |
| Hình 3.5. Đoạn code tạo form đăng ký “form-box register” | 13 |
| Hình 3.6. Đoạn code file File register.inc.php | 14 |
| Hình 3.7. Đoạn code của function để tạo user (createUser) | 15 |
| Hình 3.8. Đoạn code của function emptyInputSignUp | 15 |
| Hình 3.9. Đoạn code của function invalidUsername | 15 |
| Hình 3.10. Đoạn code của function invalidEmail | 16 |
| Hình 3.11. Đoạn code của function usernameExists | 16 |
| Hình 3.12. Đoạn code tạo file dùng để đăng xuất (file logout.inc.php) | 17 |
| Hình 3.13. Đoạn code tạo section hiển thị danh sách công việc | 17 |
| Hình 3.14. Đoạn code tạo file để lấy dữ liệu từ database | 17 |
| Hình 3.15. Đoạn code dùng để lưu lại dữ liệu | 18 |
| Hình 3.16. Cấu trúc SQL của bảng lists | 18 |
| Hình 3.17. Đoạn code tạo section dùng để quản lý tiến trình của dự án | 18 |
| Hình 3.18. Đoạn code dùng để lấy dữ liệu | 19 |
| Hình 3.19. Đoạn code dùng để lưu lại dữ liệu | 19 |
| Hình 3.20. Cấu trúc SQL của bảng kanban | 19 |
| Hình 4.1. Thanh điều hướng (Navigation Bar) | 20 |
| Hình 4.2. Giao diện trang chủ | 20 |

| | |
|--|----|
| Hình 4.3. Giao diện của thẻ About | 21 |
| Hình 4.4. Giao diện của thẻ Contact..... | 21 |
| Hình 4.5. Giao diện cá nhân của Hào | 22 |
| Hình 4.6. Giao diện cá nhân của Hào (tt) | 22 |
| Hình 4.7. Giao diện trang cá nhân của Khoa..... | 23 |
| Hình 4.8. Giao diện trang cá nhân của Khoa (tt)..... | 23 |
| Hình 4.9. Giao diện trang cá nhân của Khoa (tt)..... | 23 |
| Hình 4.10. Giao diện trang cá nhân của Nguyên..... | 24 |
| Hình 4.11. Giao diện trang cá nhân của Nguyên (tt)..... | 24 |
| Hình 4.12. Giao diện tra cá nhân của Nguyên (tt)..... | 24 |
| Hình 4.13. Giao diện trang cá nhân của My | 25 |
| Hình 4.14. Giao diện trang cá nhân của My (tt)..... | 25 |
| Hình 4.15. Giao diện trang cá nhân của My (tt)..... | 25 |
| Hình 4.16. Giao diện trang cá nhân của Thịnh..... | 26 |
| Hình 4.17. Giao diện trang cá nhân của Thịnh (tt)..... | 26 |
| Hình 4.18. Giao diện đăng nhập | 27 |
| Hình 4.19. Giao diện đăng ký..... | 27 |
| Hình 4.20. Giao diện đăng nhập | 28 |
| Hình 4.21. Thông báo nhập sai Username/Passwork | 28 |
| Hình 4.22. Giao diện trang WorkFlow (To-do List) | 29 |
| Hình 4.23. Tạo New list | 29 |
| Hình 4.24. Tạo các task cho list | 30 |
| Hình 4.25. Giao diện task-board..... | 30 |
| Hình 4.26. Tạo nhiệm vụ mới trong task-board | 31 |
| Hình 4.27. Kéo nhiệm vụ vào ô tương ứng với tiến độ làm việc thực tế | 31 |
| Hình 4.28. Thông báo xoá nhiệm vụ | 32 |
| Hình 4.29. Đăng xuất..... | 32 |

CHƯƠNG 1. GIỚI THIỆU

1.1. Lý do chọn đề tài

Việc quản lý công việc là một trong những vấn đề quan trọng và cần thiết trong quá trình làm việc của mỗi người. Song, việc quản lý thủ công trên giấy tờ có thể gây ra rất nhiều khó khăn và rủi ro trong việc quản lý và bảo mật chúng. Vì lý do trên, nhóm chúng em đã quyết định chọn đề tài “*Xây dựng trang web quản lý công việc*” để giải quyết vấn đề này.

Trang web quản lý công việc có thể giúp cho người sử dụng dễ dàng và nhanh chóng quản lý công việc của mình, bao gồm việc tạo mới, chỉnh sửa, xóa công việc và theo dõi tiến độ thực hiện. Ngoài ra, trang web còn giúp cho người dùng phân chia công việc đúng mức độ ưu tiên một cách hiệu quả.

Với sự phát triển của công nghệ thông tin, việc sử dụng trang web để quản lý công việc trở nên phổ biến hơn bao giờ hết. Vì vậy, chúng em tin rằng việc xây dựng trang web quản lý công việc sẽ mang lại nhiều lợi ích cho người sử dụng và đáp ứng được nhu cầu quản lý công việc hiện nay.

1.2. Mục tiêu của đề tài

- Xây dựng một trang web quản lý công việc dành cho cá nhân
- Triển khai chương trình

1.3. Phạm vi đề tài

Trong phạm vi của đề tài này chúng em sẽ thực hiện những công việc sau:

- Tìm hiểu về việc quản lý công việc theo hướng thủ công
- Ứng dụng được xây dựng bằng phần mềm Visual Studio Code và Xampp
- Sử dụng các ngôn ngữ lập trình như html, css, javascript, PHP và xây dựng cơ sở dữ liệu bằng ứng dụng MySQL.

1.4. Phương thức thực hiện

Để thực hiện đề tài này, em sử dụng phương pháp phân tích và thiết kế hướng đối tượng có cấu trúc – bao gồm các hoạt động khảo sát, phân tích, thiết kế, bổ sung, kiểm thử chất lượng... các hoạt động có thể thực hiện song song với nhau.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Lý thuyết cơ sở dữ liệu

2.1.1. Khái niệm

Cơ sở dữ liệu (viết tắt CSDL; tiếng Anh là database) là một tập hợp dữ liệu được tổ chức theo một cấu trúc cụ thể để lưu trữ và quản lý thông tin. Các dữ liệu này được tổ chức thành các bảng (tables) và các quan hệ giữa chúng được xác định bằng các khóa chính (primary keys) và các khóa ngoại (foreign keys).



Hình 2.1. Cơ sở dữ liệu (database)

Cơ sở dữ liệu được sử dụng để lưu trữ và quản lý thông tin trong các hệ thống thông tin, từ các ứng dụng cá nhân cho đến các hệ thống doanh nghiệp lớn. Các cơ sở dữ liệu có thể được quản lý và truy cập bằng các hệ quản trị cơ sở dữ liệu (Database Management System - DBMS) như MySQL, Oracle, SQL Server, PostgreSQL, MongoDB, ...

Sự phát triển của công nghệ cơ sở dữ liệu đã tạo ra nhiều khả năng và tiện ích cho việc lưu trữ và quản lý thông tin, cải thiện hiệu quả và độ chính xác của quản lý thông tin. Cơ sở dữ liệu cũng đóng vai trò quan trọng trong việc tích hợp các hệ thống khác nhau và cung cấp các dịch vụ trực tuyến phục vụ cho người dùng.

2.1.2. Ưu điểm

- Giảm sự trùng lặp thông tin xuống mức tối thiểu nhất. Do đó đảm bảo thông tin có tính nhất quán và toàn vẹn dữ liệu.
- Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.
- Nhiều người có thể sử dụng một cơ sở dữ liệu.

2.1.3. Nhược điểm

- *Tính chủ quyền của dữ liệu:*

Thể hiện ở phương diện an toàn dữ liệu. Khả năng biểu diễn mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu. Người khai thác cơ sở dữ liệu phải cập nhật cho CSDL những thông tin mới nhất.

- *Tính bảo mật và quyền khai thác thông tin của người sử dụng:*

Do ưu điểm CSDL có thể cho nhiều người khai thác đồng thời, nên cần phải có một cơ chế bảo mật phân quyền khai thác CSDL. Các hệ điều hành nhiều người sử dụng hay cục bộ đều cung cấp cơ chế này.

- *Tranh chấp dữ liệu:*

Khi nhiều người cùng truy nhập CSDL với các mục đích khác nhau. Rất có thể sẽ xảy ra hiện tượng tranh chấp dữ liệu. Cần có cơ chế ưu tiên khi truy cập CSDL. Ví dụ: admin có thể truy cập cơ sở dữ liệu. Cấp quyền ưu tiên cho từng người khai thác.

- *Cần đảm bảo an toàn dữ liệu khi có sự cố:*

Khi CSDL nhiều và được quản lý tập trung. Khả năng rủi ro mất dữ liệu rất cao. Các nguyên nhân chính là mất điện đột ngột hoặc hỏng thiết bị lưu trữ. Hiện tại có một số hệ điều hành đã có cơ chế tự động sao lưu ổ cứng và fix lỗi khi có sự cố xảy ra. Vì vậy, chúng ta nên sao lưu cho dữ liệu để phòng trường hợp xấu xảy ra.

2.1.4. Phân loại cơ sở dữ liệu

Có 4 cách phân loại cơ sở dữ liệu như sau:

- Phân loại theo loại dữ liệu: Cơ sở dữ liệu có cấu trúc; Cơ sở dữ liệu phi cấu trúc; Cơ sở dữ liệu bán cấu trúc.

- Phân loại theo hình thức lưu trữ, mô hình tổ chức: Cơ sở dữ liệu dạng tệp; Cơ sở dữ liệu quan hệ; Cơ sở dữ liệu phân cấp.
- Phân loại theo đặc tính sử dụng: Cơ sở dữ liệu kho; Cơ sở dữ liệu ngữ nghĩa; Cơ sở dữ liệu hoạt động.
- Phân loại theo mô hình triển khai: Cơ sở dữ liệu tập trung; Cơ sở dữ liệu phân tán; Cơ sở dữ liệu tập trung có bản sao.

2.2. Hệ quản trị cơ sở dữ liệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở được sử dụng rộng rãi trên toàn thế giới. MySQL cung cấp nhiều tính năng nâng cao, công cụ quản lý và hỗ trợ kỹ thuật để đạt được mức độ cao nhất về khả năng mở rộng, bảo mật, độ tin cậy và thời gian hoạt động của MySQL.



Hình 2.2. MySQL là gì?

Ưu điểm của MySQL là có thể chạy trên mọi ngôn ngữ như ASP, .NET, PHP, JSP, ... Khi chạy không đòi hỏi tài nguyên máy quá lớn như CPU, RAM, ...

Nhược điểm của MySQL là bảo mật chưa cao và quá trình restore có phần hơn chậm hơn so với một số hệ quản trị cơ sở dữ liệu khác.

2.3. Ngôn ngữ lập trình xây dựng hệ thống

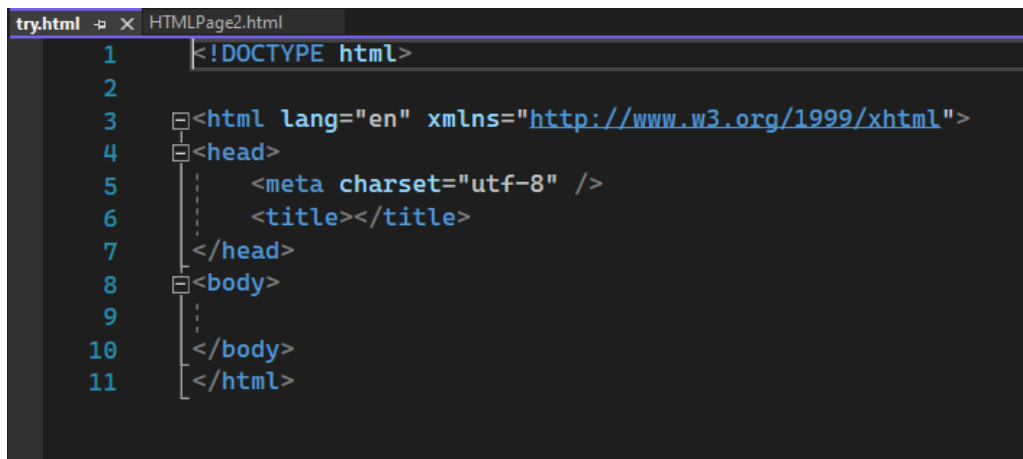
2.3.1. HTML và CSS

- *HTML (HyperText Markup Language)*

Được sử dụng để tạo và cấu trúc các thành phần trong trang một trang web, HTML không phải là ngôn ngữ lập trình, HTML sử dụng các thẻ(tag) để cấu trúc nội dung của trang web.

Cách hoạt động HTML: muốn chạy được file HTML ta phải nhờ vào trình duyệt, trình duyệt sẽ đóng vai trò đọc các file HTML và xuất ra nội dung trên internet sao cho người dùng có thể đọc hiểu được.

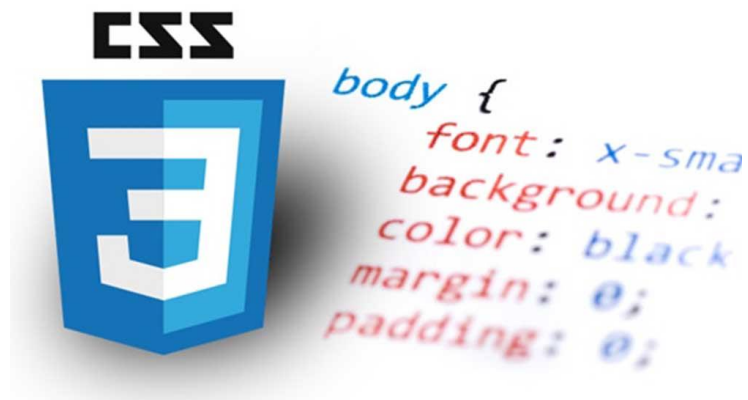
Bố cục cơ bản của một trang web HTML:



Hình 2.3. Bố cục cơ bản của một trang HTML

- <!DOCTYPE html>: khai báo kiểu dữ liệu hiển thị.
 - <html>: xác định một tài liệu html, là cặp thẻ bắt buộc.
 - <head>: chứa các thẻ cung cấp thông tin về trang web như: tiêu đề, charset, ...
 - <title>: cặp thẻ nằm trong thẻ head, khai báo tiêu đề của trang (hiển thị trên thanh tiêu đề của browser)
 - <body>: chứa tất cả các nội dung hiển thị trên trang web như văn bản, hình ảnh, video, ...
- **CSS (Cascading Style Sheets)**

Là ngôn ngữ dùng để hoàn thiện trang web HTML, giúp chúng ta trình bày bố cục của trang web. CSS giúp người dùng nhiều styles trên một trang web HTML nên khả năng điều chỉnh trang của bạn trở nên vô hạn.

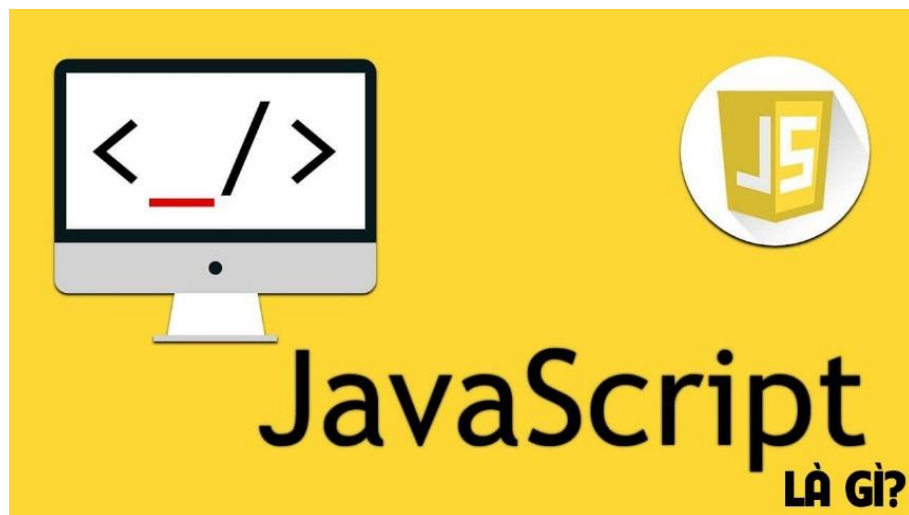


Hình 2.4. Ngôn ngữ CSS

2.3.2. Ngôn ngữ lập trình JavaScript và PHP

- **JavaScript**

Là một ngôn ngữ lập trình web rất phổ biến ngày nay. Javascript được tích hợp đồng thời nhúng vào HTML để hỗ trợ cho website trở nên sống động hơn. Chúng cũng đóng vai trò tương tự như một phần của website, cho phép Client-side Script từ người dùng tương tự máy chủ (Nodejs) để tạo ra những website động. Tất cả những đoạn mã JS đều sẽ được đặt ở trong cặp thẻ đóng và mở `<script></script>`



Hình 2.5. JavaScript là gì?

Cách hoạt động:

- Thông thường, JavaScript sẽ được nhúng trực tiếp vào một website hoặc chúng được tham chiếu qua file .js hoặc .JavaScript.

- Đây là một ngôn ngữ đến từ phía Client nên Script sẽ được download về máy client khi truy cập.

- Tại đây, chúng sẽ được hệ thống xử lý. Vì vậy, bạn không cần phải tải về máy server rồi chờ cho chúng xử lý xong mới phản hồi được kết quả đến client.

- **PHP (Hypertext Preprocessor)**

Đây là một ngôn ngữ lập trình kịch bản (scripting language) đa mục đích. PHP được dùng phổ biến cho việc phát triển các ứng dụng web chạy trên máy chủ. Do đó, ngôn ngữ lập trình PHP có thể xử lý các chức năng từ phía server để sinh ra mã HTML trên client như thu thập dữ liệu biểu mẫu, sửa đổi cơ sở dữ liệu, quản lý file trên server hay các hoạt động khác.



Hình 2.6. Ngôn ngữ PHP

Hiện nay rất phổ biến việc sử dụng mã nguồn mở để nhúng vào trong HTML nhờ sử dụng cặp thẻ PHP khi áp dụng trong việc tối ưu hóa cho các ứng dụng web đã đem lại những hiệu quả cao như tốc độ xử lý nhanh, tiện lợi, ít lỗi mà cấu trúc tương tự như Java hay C. Ngoài ra, PHP cũng được coi là một ngôn ngữ khá dễ học và thành thạo hơn so với các ngôn ngữ khác. Vì vậy đây là lý do mà PHP ngày càng được áp dụng nhiều và trở thành ngôn ngữ lập trình phổ biến nhất.

Ngoài ra, PHP còn cung cấp các tính năng rất mạnh mẽ cho những người lập trình chuyên nghiệp.

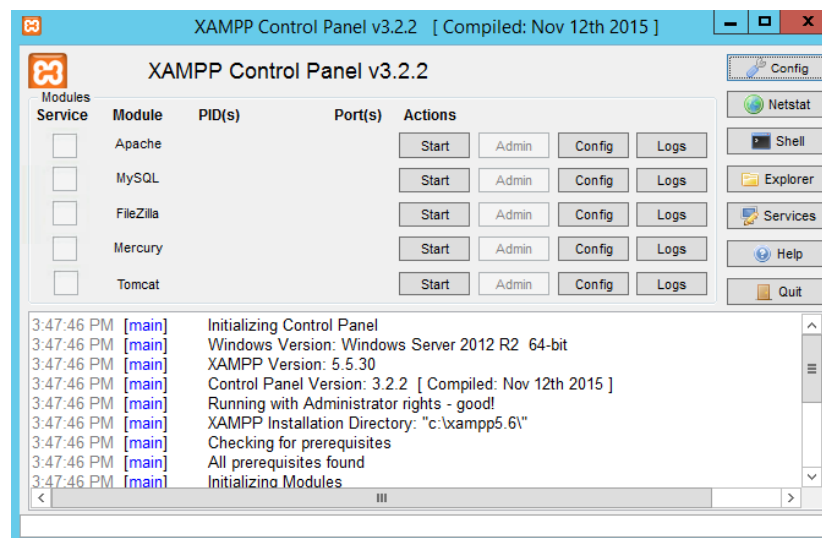
Mặc dù sở hữu được nhiều lợi ích nhưng PHP vẫn có một số hạn chế nhất định như vấn đề bảo mật chưa cao do bản chất của PHP có mã nguồn mở nên có nhiều lỗ

hông; PHP chỉ hoạt động được trên các website và giao diện không được gọn gàng, đẹp mắt. Độ bảo mật và hiệu suất của ngôn ngữ này cũng chưa tốt.

2.4. XAMPP

XAMPP là một loại phần mềm khá phổ biến và thường được các lập trình viên sử dụng để xây dựng và phát triển các dự án website theo ngôn ngữ PHP. XAMPP thường được dùng để tạo máy chủ web được tích hợp sẵn PHP, MySQL, Apache, FTP Server, Mail Server và các công cụ như phpMyAdmin.

XAMPP hoạt động dựa trên sự tích hợp của 5 phần mềm chính là Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) và Perl (P), nên tên gọi XAMPP cũng là viết tắt từ chữ cái đầu của 5 phần mềm này.



Hình 2.7. Giao diện chính của Xampp

Các ưu điểm của XAMPP bao gồm:

- XAMPP có thể chạy được trên toàn bộ các hệ điều hành như Window, MacOS, Cross-platform và Linux.
- XAMPP có cấu hình đơn giản cùng như nhiều chức năng hữu ích cho người dùng. Tiêu biểu là giải lập Server, giả lập Mail Server, hỗ trợ SSL trên Localhost.
- XAMPP tích hợp nhiều thành phần với các chức năng Apache, PHP hay MySql...

- Mã nguồn mở: Không như Appserv, XAMPP có giao diện quản lý khá tiện lợi. Nhờ đó, người dùng có thể chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.

Tuy nhiên, XAMPP cũng có một số nhược điểm sau:

- Do cấu hình đơn giản nên XAMPP không được hỗ trợ cấu hình Module nên cũng không có Version MySQL. Do đó đôi khi sẽ mang đến sự bất tiện cho từng người. Trong khi WAMP có nhiều tùy chọn hơn vì nó có nhiều phiên bản cho từng thành phần của server như PHP, Apache, MySQL.

- Dung lượng của XAMPP cũng tương đối nặng, dung lượng file cài đặt của XAMPP là 141Mb, nặng hơn nhiều so với WAMP chỉ 41Mb.

CHƯƠNG 3. CHỨC NĂNG CỦA HỆ THỐNG

3.1. Chức năng đăng nhập

Tại file index.php tạo form đăng nhập “form-box login” các thông tin bao gồm username và password.

```
<div class="wrapper">
  <div class="form-box login">
    <span class="icon-close login-close"><ion-icon name="close"></ion-icon></span>
    <h2>Login</h2>
    <form action="../../PHP/includes/login.inc.php" method="post">
      <div class="input-box">
        <span class="icon"><ion-icon name="person"></ion-icon></span>
        <input type="text" name="username" required />
        <label>Username/Email</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
        <input type="password" name="password" required />
        <label>Password</label>
      </div>
      <button type="submit" class="btn" name="submit">Login</button>
      <div class="login-register">
        <p>
          Don't have an account?
          <a href="#" class="register-link">Register</a>
        </p>
      </div>
    </form>
  </div>
```

Hình 3.1. Đoạn code tạo form đăng nhập “form-box login”

Tạo file functions.inc.php viết functionlogin user.

```
function loginUser($conn, $username, $password)
{
    $usernameExists = usernameExists($conn, $username, $username);

    if ($usernameExists === false) {
        header("location: ../../HTML/index.php?error=wronglogin");
        exit();
    }

    $passwordHashed = $usernameExists["usersPwd"];
    $checkPassword = password_verify($password, $passwordHashed);

    if ($checkPassword === false) {
        header("location: ../../HTML/index.php?error=wronglogin");
        exit();
    } else if ($checkPassword === true) {
        session_start();
        $_SESSION["userid"] = $usernameExists["usersId"];
        $_SESSION["username"] = $usernameExists["usersName"];
        header("location: ../../HTML/work.php");
        exit();
    }
}
```

Hình 3.2. Đoạn code function loginUser

File login.inc.php: khi nhập thông tin vào 2 ô username và password thì sẽ post thông tin vào login.inc.php, emptyInputLogin kiểm tra dữ liệu nhập có trống hay không, nếu trống thì báo lỗi, còn nếu điền đầy đủ các thông tin thì chạy loginUser nằm trong file function.inc.php.

```
<?php
if (isset($_POST["submit"])) {
    $username = $_POST["username"];
    $password = $_POST["password"];

    require_once 'dbh.inc.php';
    require_once 'functions.inc.php';

    if (emptyInputLogin($username, $password) !== false) {
        header("location: ../../HTML/index.php?error=emptyinput");
        exit();
    }

    loginUser($conn, $username, $password);
} else {
    header("location: ../../HTML/index.php");
    exit();
}
```

Hình 3.3. Đoạn code file login.inc.php

Cấu trúc SQL như sau:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|--------------------------|------|-----------|--------------|--------------------|------|---------|----------|----------------|--|
| <input type="checkbox"/> | 1 | usersId | int(11) | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> | 2 | usersName | varchar(128) | utf8mb4_unicode_ci | No | None | | | Change Drop More |
| <input type="checkbox"/> | 3 | userEmail | varchar(128) | utf8mb4_unicode_ci | No | None | | | Change Drop More |
| <input type="checkbox"/> | 4 | usersPwd | varchar(128) | utf8mb4_unicode_ci | No | None | | | Change Drop More |

Hình 3.4. Cấu trúc SQL của bảng users

3.2. Chức năng đăng kí

Tương tự như đăng nhập, tạo “form-box register” trong file index.php.

```

<div class="form-box register">
  <span class="icon-close register-close"><ion-icon name="close"></ion-icon></span>
  <h2>Registration</h2>
  <form action="../PHP/includes/register.inc.php" method="post">
    <div class="input-box">
      <span class="icon"><ion-icon name="person"></ion-icon></span>
      <input type="text" name="username" required />
      <label>Username</label>
    </div>
    <div class="input-box">
      <span class="icon"><ion-icon name="mail"></ion-icon></span>
      <input type="email" name="email" required />
      <label>Email</label>
    </div>
    <div class="input-box">
      <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
      <input type="password" name="password" required />
      <label>Password</label>
    </div>
    <button type="submit" class="btn" name="submit">Register</button>
    <div class="login-register">
      <p>
        Already have an account?
        <a href="#" class="login-link">Login</a>
      </p>
    </div>
  </form>
</div>
</div>

```

Hình 3.5. Đoạn code tạo form đăng ký "form-box register"

File register.inc.php: khi điền đủ thông tin thì chạy function createUser. Ngược lại nếu không thông tin nào thì chạy function emptyInputSignUp hoặc một trong các thông tin đã tồn tại thì chạy function invalidUsername, invalidEmail, usernameExists.

```
<?php

if (isset($_POST["submit"])) {

    $username = $_POST["username"];
    $email = $_POST["email"];
    $password = $_POST["password"];

    require_once 'dbh.inc.php';
    require_once 'functions.inc.php';

    if (emptyInputSignUp($username, $email, $password) !== false) {
        header("location: ../../HTML/index.php?error=emptyinput");
        exit();
    }

    if (invalidUsername($username) !== false) {
        header("location: ../../HTML/index.php?error=invalidusername");
        exit();
    }

    if (invalidEmail($email) !== false) {
        header("location: ../../HTML/index.php?error=invalidemail");
        exit();
    }

    if (usernameExists($conn, $username, $email) !== false) {
        header("location: ../../HTML/index.php?error=usernamealreadytaken");
        exit();
    }

    createUser($conn, $username, $email, $password);
} else {
    header("location: ../../HTML/index.php");
    exit();
}
```

Hình 3.6. Đoạn code file File register.inc.php

Function createUser:

```
function createUser($conn, $username, $email, $password)
{
    $sql = "INSERT INTO users (userName, userEmail, usersPwd) VALUES (?, ?, ?);";
    $stmt = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmt, $sql)) {
        header("location: ../../HTML/index.php?error=stmtfailed");
        exit();
    }

    $hashPassword = password_hash($password, PASSWORD_DEFAULT);

    mysqli_stmt_bind_param($stmt, "sss", $username, $email, $hashPassword);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
    header("location: ../../HTML/index.php?error=none");
    exit();
}
```

Hình 3.7. Đoạn code của function để tạo user (createUser)

Function emptyInputSignUp:

```
function emptyInputSignUp($username, $email, $password)
{
    $result = false;
    if (empty($username) || empty($email) || empty($password)) {
        $result = true;
    }
    return $result;
}
```

Hình 3.8. Đoạn code của function emptyInputSignUp

Function invalidUsername:

```
function invalidUsername($username)
{
    $result = false;
    if (!preg_match("/^[a-zA-Z0-9]*$/", $username)) {
        $result = true;
    }
    return $result;
}
```

Hình 3.9. Đoạn code của function invalidUsername

Function invalidEmail:

```
function invalidEmail($email)
{
    $result = false;
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $result = true;
    }
    return $result;
}
```

Hình 3.10. Đoạn code của function invalidEmail

Function usernameExists:

```
function usernameExists($conn, $username, $email)
{
    $sql = "SELECT * FROM users WHERE userName = ? OR userEmail = ?";
    $stmt = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmt, $sql)) {
        header("location: ../../HTML/index.php?error=stmtfailed");
        exit();
    }

    mysqli_stmt_bind_param($stmt, "ss", $username, $email);
    mysqli_stmt_execute($stmt);

    $resultData = mysqli_stmt_get_result($stmt);

    if ($row = mysqli_fetch_assoc($resultData)) {
        return $row;
    } else {
        $result = false;
        return $result;
    }

    mysqli_stmt_close($stmt);
}
```

Hình 3.11. Đoạn code của function usernameExists

3.3. Chức năng đăng xuất

Tạo file logout.inc.php.

```
<?php
session_start();
session_unset();
session_destroy();
header("location: ../../HTML/index.php");
exit();
```

Hình 3.12. Đoạn code tạo file dùng để đăng xuất (file logout.inc.php)

3.4. Todo-list

Dùng để hiển thị danh sách các công việc.

```
<!--To-Do List Section-->
<section id="todolist-section" class="showsection">
  <div class="todogrid">
    <div class="all-tasks">
      <form action="" data-new-list-form>
        <input type="text" class="new list" data-new-list-input placeholder="New list name" aria-label="new list name" />
        <button class="btn create" aria-label="create new list">+</button>
      </form>
    </div>
    <div id="todo-list-display-container" data-lists></div>
  </div>
</section>
```

Hình 3.13. Đoạn code tạo section hiển thị danh sách công việc

Tạo file todo_getData.php để lấy dữ liệu từ database qlcongviec trong sql đã lưu lại từ các thao tác mà người dùng đã thực hiện từ lần đăng nhập trước.

```
<?php

session_start();

$serverName = "localhost";
$dbUserName = "root";
$dbPassword = "";
$dbName = "qlcongviec";

$conn = mysqli_connect($serverName, $dbUserName, $dbPassword, $dbName);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$userId = $_SESSION["userid"];
$result = mysqli_query($conn, "SELECT * FROM lists WHERE usersId = '" . $userId . "'");
$row = mysqli_fetch_assoc($result);

echo trim(stripslashes(json_encode($row["listsData"])), '');
```

Hình 3.14. Đoạn code tạo file để lấy dữ liệu từ database

File `todo_saveData.php`: lưu lại dữ liệu mà người dùng thực hiện khi đang đăng nhập.

```
<?php
session_start();

$data = $_POST['data'];

$serverName = "localhost";
$dbUserName = "root";
$dbPassword = "";
$dbName = "qlcongviiec";

$conn = mysqli_connect($serverName, $dbUserName, $dbPassword, $dbName);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$userId = $_SESSION["userid"];
$result = mysqli_query($conn, "UPDATE lists SET listsData = '" . $data . "' WHERE usersId = '" . $userId . "'");
```

Hình 3.15. Đoạn code dùng để lưu lại dữ liệu

Cấu trúc SQL như sau:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|--------------------------|------|-----------|---------------|--------------------|------|---------|----------|----------------|--|
| <input type="checkbox"/> | 1 | listsId | int(11) | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> | 2 | usersId | int(11) | | No | None | | | Change Drop More |
| <input type="checkbox"/> | 3 | listsData | varchar(5000) | utf8mb4_unicode_ci | No | [] | | | Change Drop More |

Hình 3.16. Cấu trúc SQL của bảng lists

3.5. Task-board

Dùng để quản lý tiến trình của dự án.

```
<section id="task-board" class="showsection" style="display: block;">
  <div class="kanban">
    <div class="kanban__column" data-id="1">
      <div class="kanban__column-title">Not Started</div>
      <div class="kanban__column-items">
        <div class="kanban__dropzone"></div>
      </div>
      <button class="kanban__add-item" type="button">+ Add</button>
    </div>
    <div class="kanban__column" data-id="2">
      <div class="kanban__column-title">In Progress</div>
      <div class="kanban__column-items">
        <div class="kanban__column-items"></div>
      </div>
      <button class="kanban__add-item" type="button">+ Add</button>
    </div>
    <div class="kanban__column" data-id="3">
      <div class="kanban__column-title">Completed</div>
      <div class="kanban__column-items">
        <div class="kanban__column-items"></div>
      </div>
      <button class="kanban__add-item" type="button">+ Add</button>
    </div>
  </div>
</section>
```

Hình 3.17. Đoạn code tạo section dùng để quản lý tiến trình của dự án

Tạo file `kanban_getData.php`: tương tự như `todo_getData` lấy dữ liệu từ trong database `qlcongviec` mà người dùng đã lưu từ trong lần đăng nhập trước.

```
<?php

session_start();

$serverName = "localhost";
$dbUserName = "root";
$dbPassword = "";
$dbName = "qlcongviec";

$conn = mysqli_connect($serverName, $dbUserName, $dbPassword, $dbName);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$userId = $_SESSION["userId"];
$result = mysqli_query($conn, "SELECT * FROM kanban WHERE usersId = '" . $userId . "'");
$row = mysqli_fetch_assoc($result);

echo trim(stripslashes(json_encode($row["kanbanData"])), '');
```

Hình 3.18. Đoạn code dùng để lấy dữ liệu

File `kanban_saveData.php`: lưu lại dữ liệu trong `kanban` mà người dùng thay đổi trang lần đăng nhập hiện tại,

```
<?php

session_start();

$data = $_POST['data'];

$serverName = "localhost";
$dbUserName = "root";
$dbPassword = "";
$dbName = "qlcongviec";

$conn = mysqli_connect($serverName, $dbUserName, $dbPassword, $dbName);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$userId = $_SESSION["userId"];
$result = mysqli_query($conn, "UPDATE kanban SET kanbanData = '" . $data . "' WHERE usersId = '" . $userId . "'");
```

Hình 3.19. Đoạn code dùng để lưu lại dữ liệu

Cấu trúc SQL như sau:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|--------------------------|------|------------|---------------|--------------------|------|---|----------|----------------|--|
| <input type="checkbox"/> | 1 | kanbanId | int(11) | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> | 2 | usersId | int(11) | | No | None | | | Change Drop More |
| <input type="checkbox"/> | 3 | kanbanData | varchar(5000) | utf8mb4_unicode_ci | No | [{"id":1,"items":[]}, {"id":2,"items":[]}, {"id":3,"items":[]}] | | | Change Drop More |

Hình 3.20. Cấu trúc SQL của bảng `kanban`

CHƯƠNG 4. GIAO DIỆN CỦA HỆ THỐNG

4.1. Trang chủ

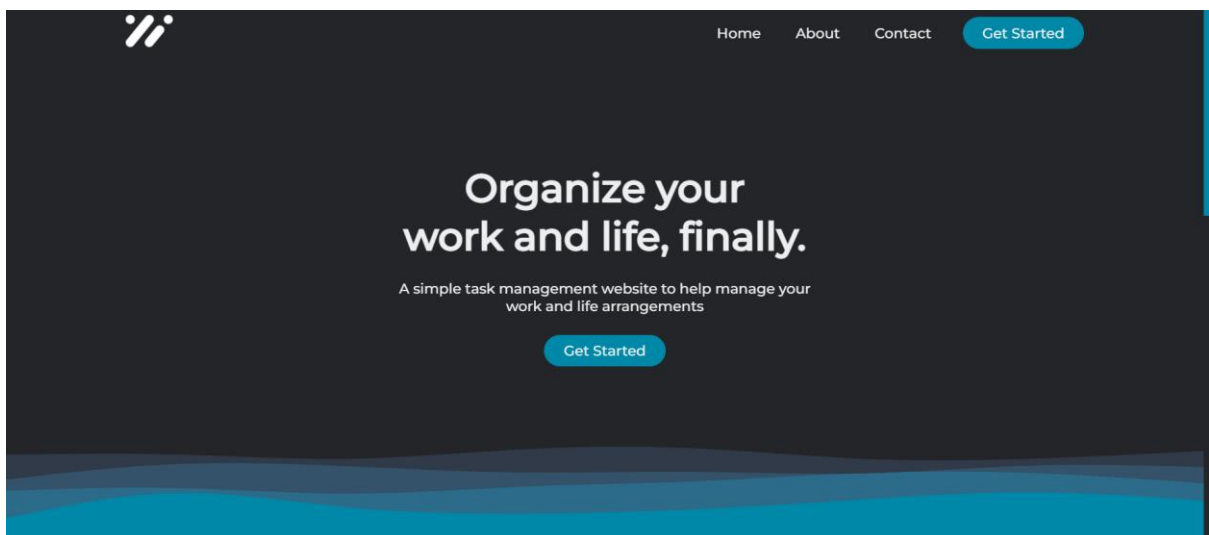
Trang chủ bao gồm:

- Thanh điều hướng (Navigation Bar) chứa các liên kết đến các thẻ Home (trang chủ), thẻ About (thẻ thông tin về trang web), thẻ Contact (thẻ thông tin của các thành viên trong nhóm), Get Started (tương tự như nút đăng nhập của trang web) và logo chính của web.



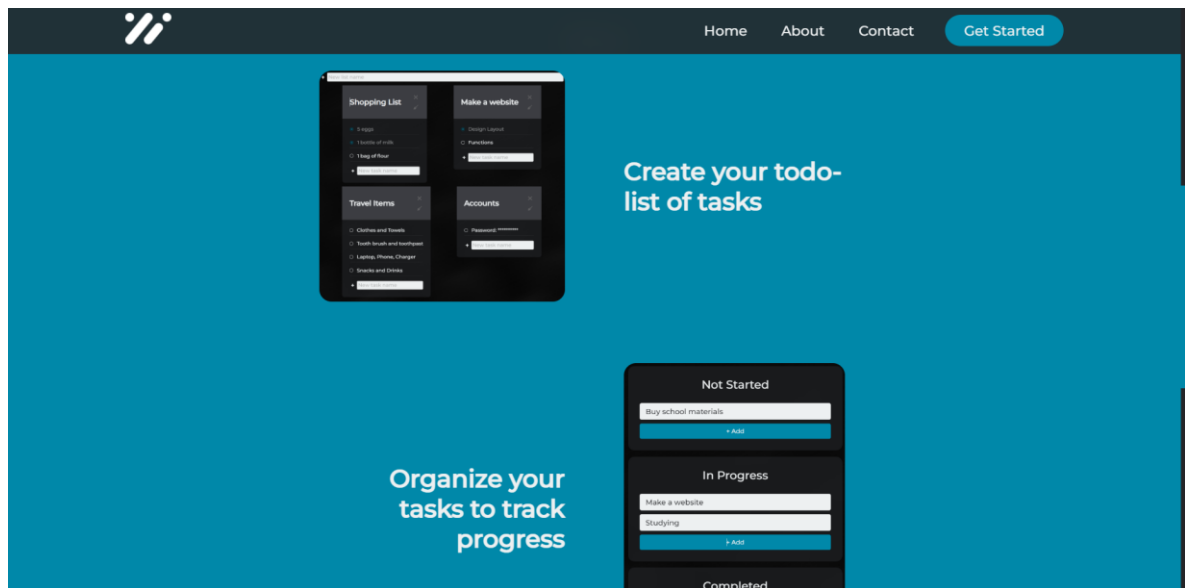
Hình 4.1. Thanh điều hướng (Navigation Bar)

- Thẻ Home: hiển thị các thông tin tổng quan về trang web.



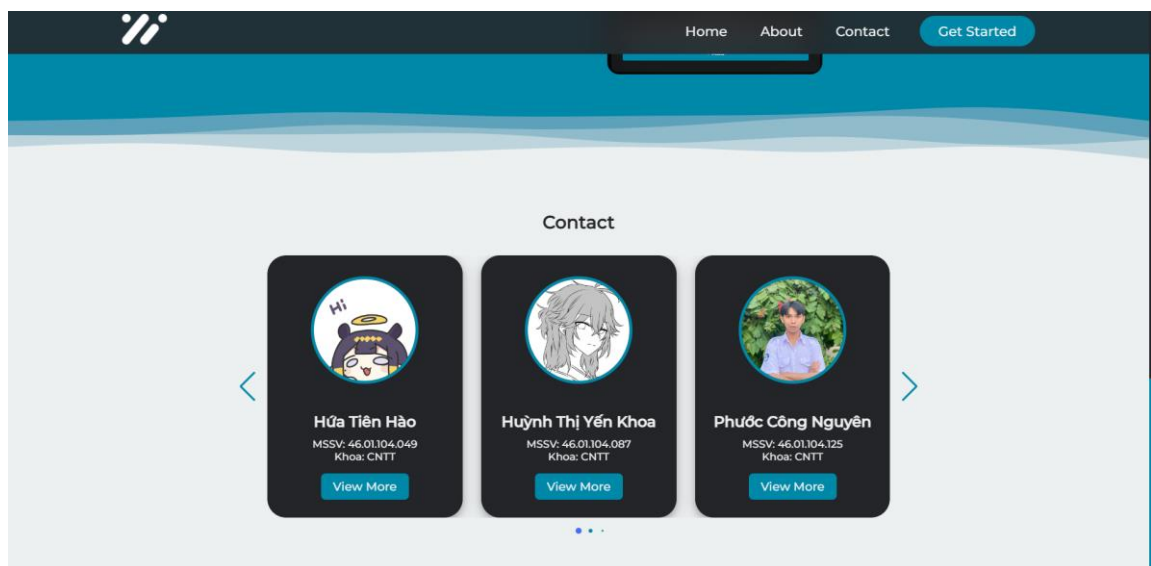
Hình 4.2. Giao diện trang chủ

- Thẻ About: hiển thị các thông tin chi tiết của trang web.



Hình 4.3. Giao diện của thẻ About

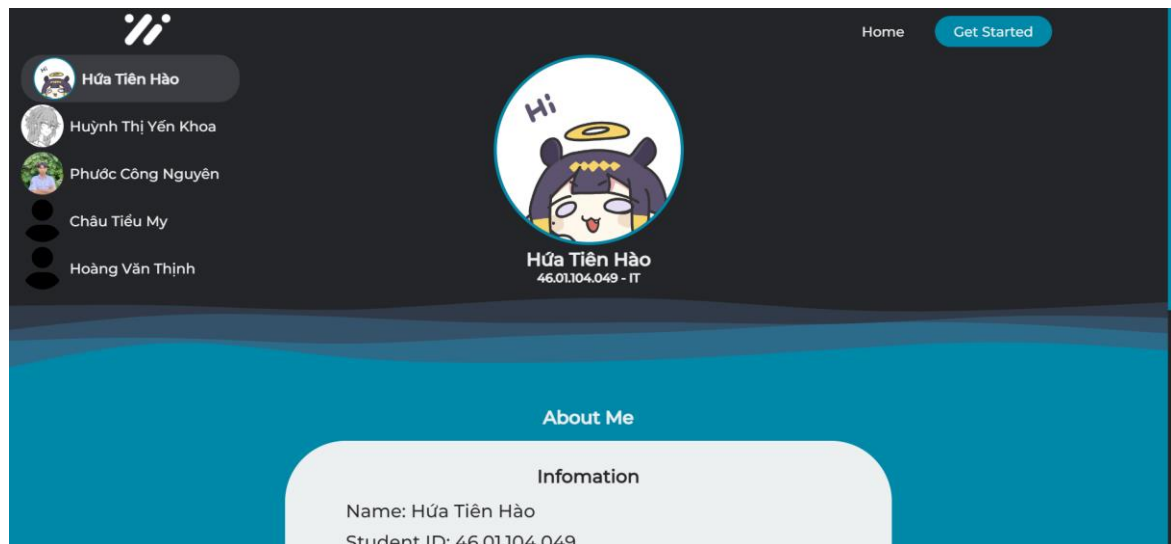
- Thẻ Contact: hiển thị các thông tin của các thành viên trong nhóm.



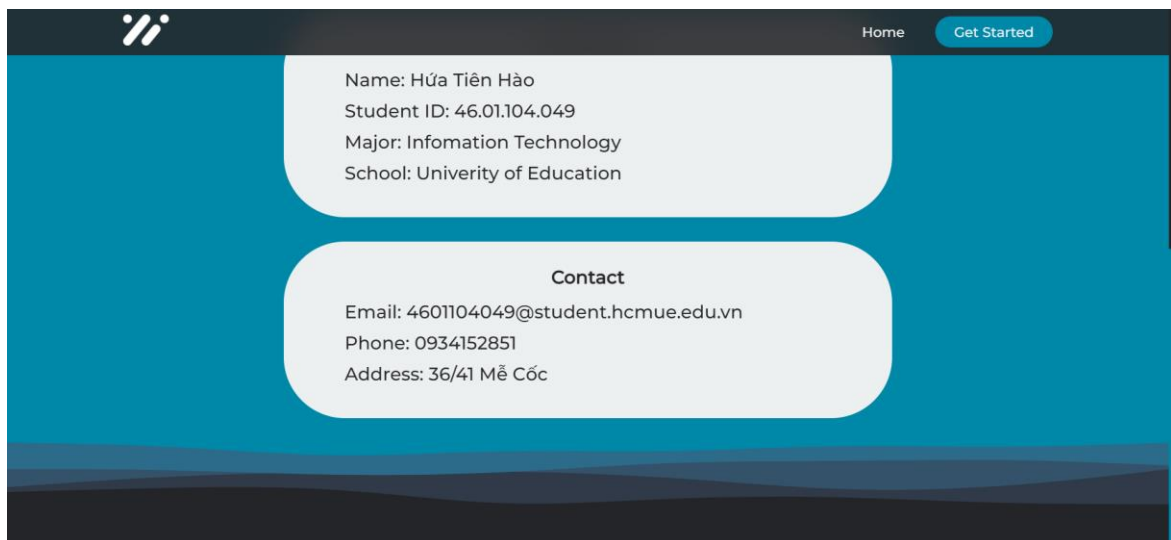
Hình 4.4. Giao diện của thẻ Contact

Để xem thông tin cá nhân của các thành viên trong nhóm, ta nhấn vào **View More** ở ô tương ứng với thành viên đó.

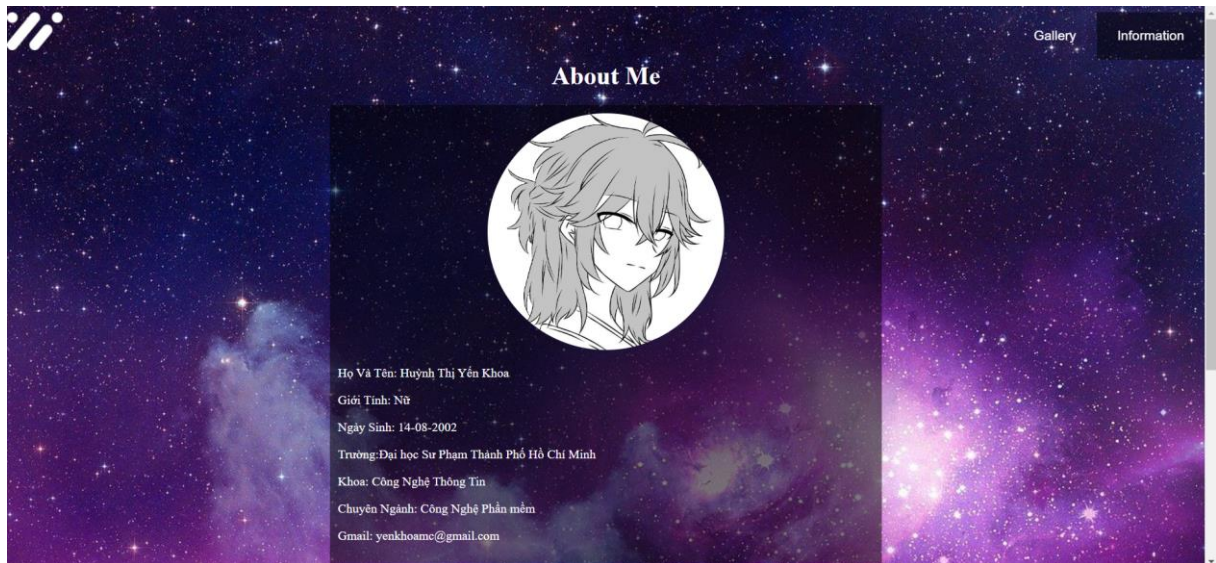
Một số giao diện thông tin cá nhân của các thành viên trong nhóm như sau:



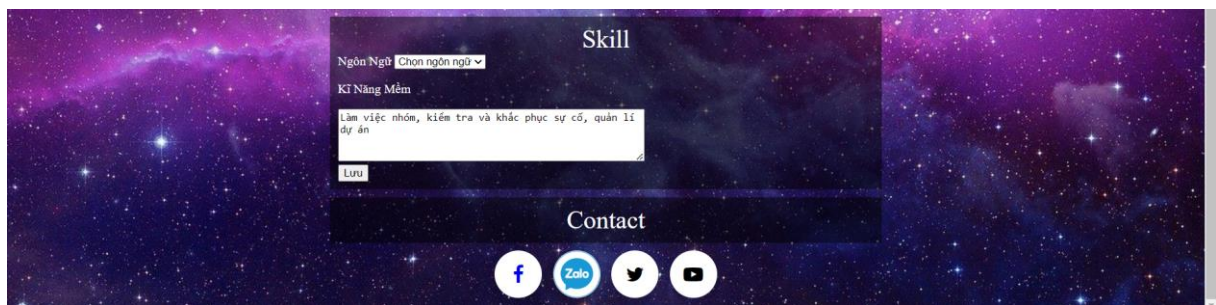
Hình 4.5. Giao diện cá nhân của Hào



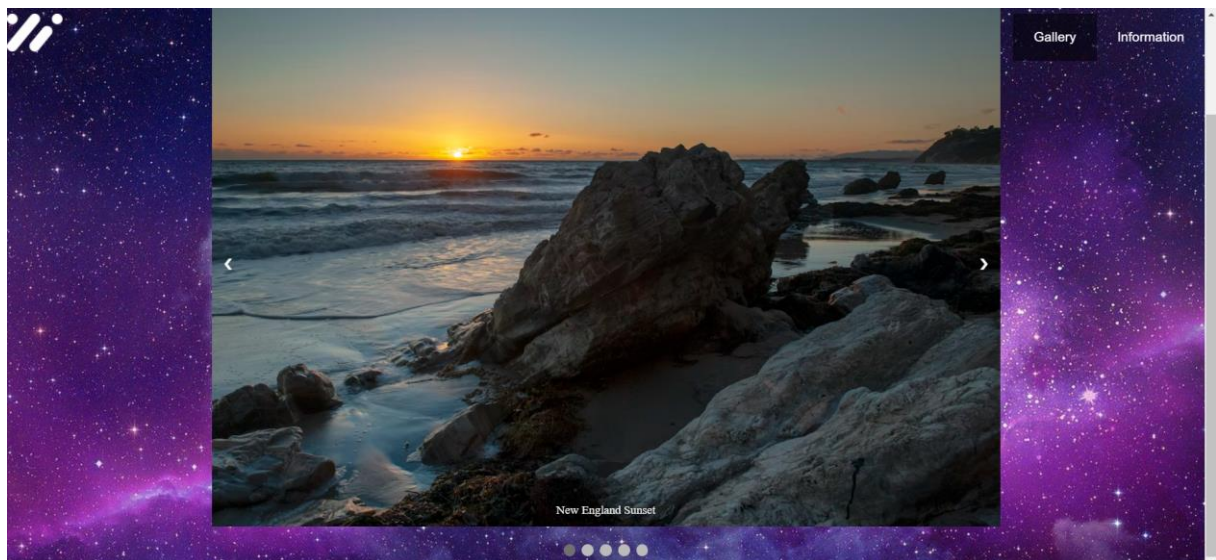
Hình 4.6. Giao diện cá nhân của Hào (tt)



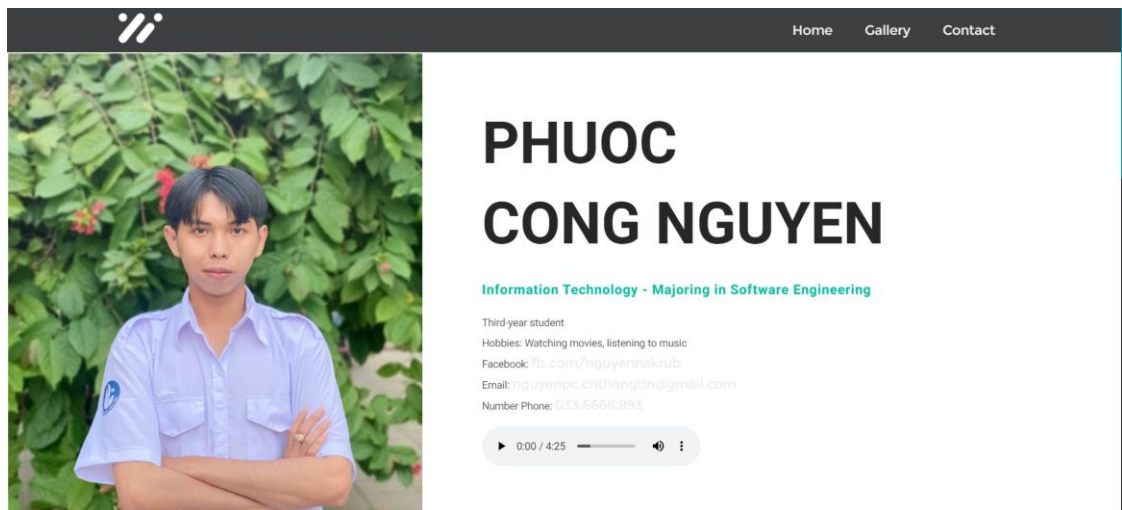
Hình 4.7. Giao diện trang cá nhân của Khoa



Hình 4.8. Giao diện trang cá nhân của Khoa (tt)



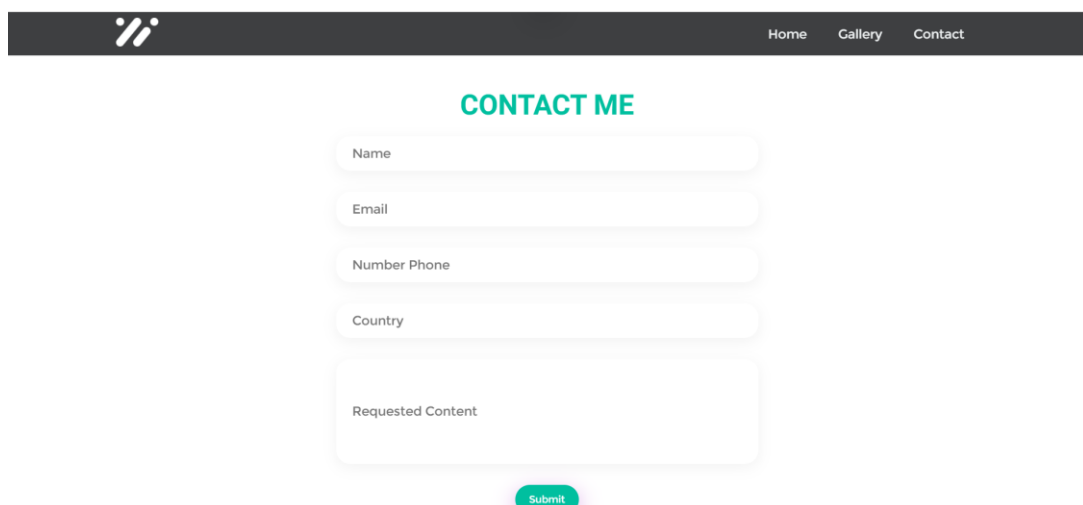
Hình 4.9. Giao diện trang cá nhân của Khoa (tt)



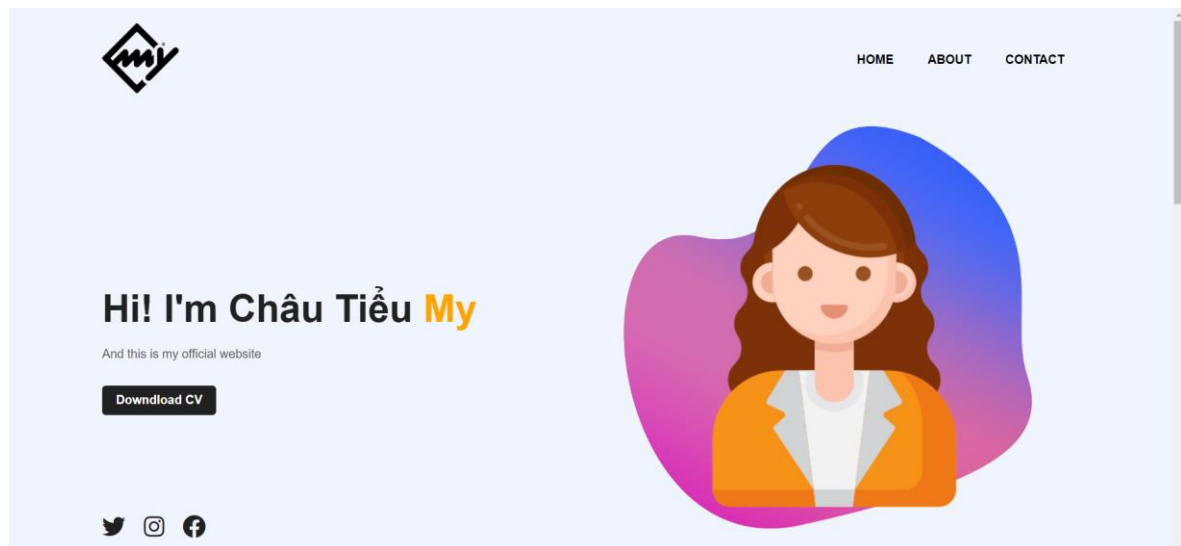
Hình 4.10. Giao diện trang cá nhân của Nguyễn



Hình 4.11. Giao diện trang cá nhân của Nguyễn (tt)



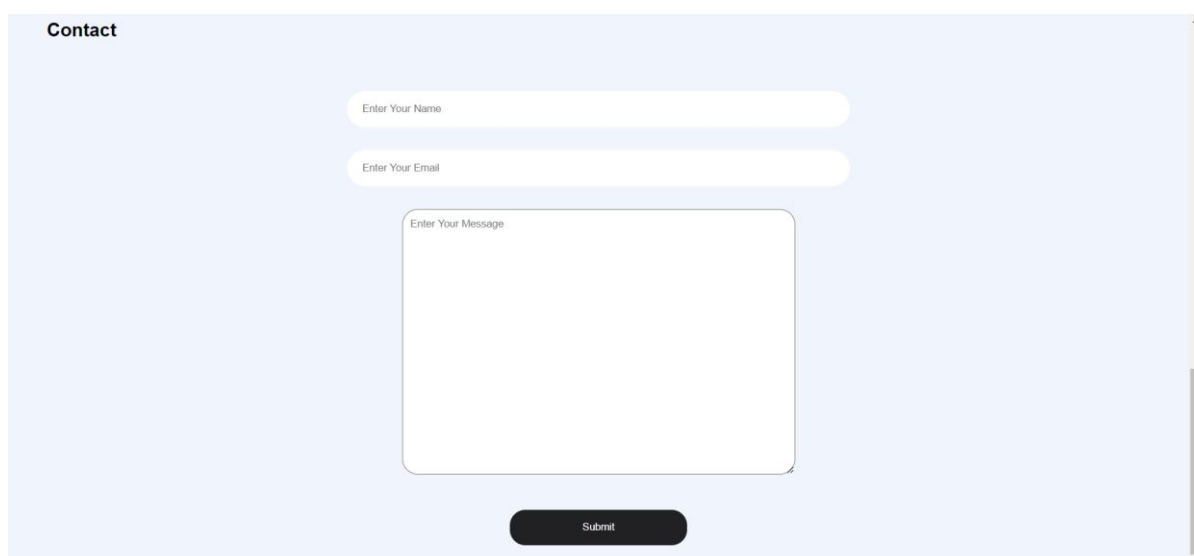
Hình 4.12. Giao diện tra cá nhân của Nguyễn (tt)



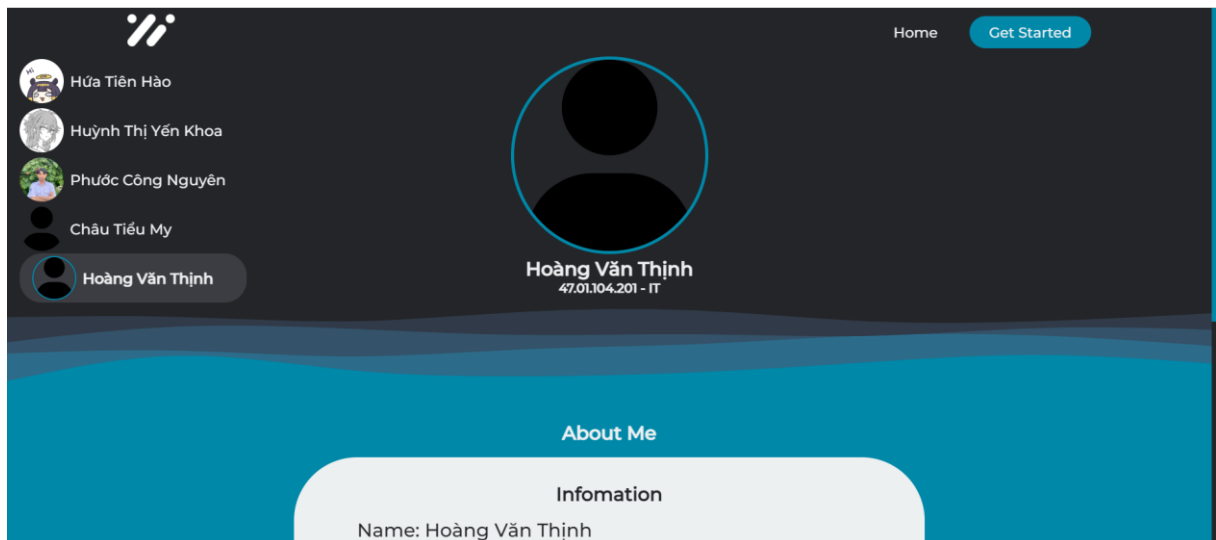
Hình 4.13. Giao diện trang cá nhân của My



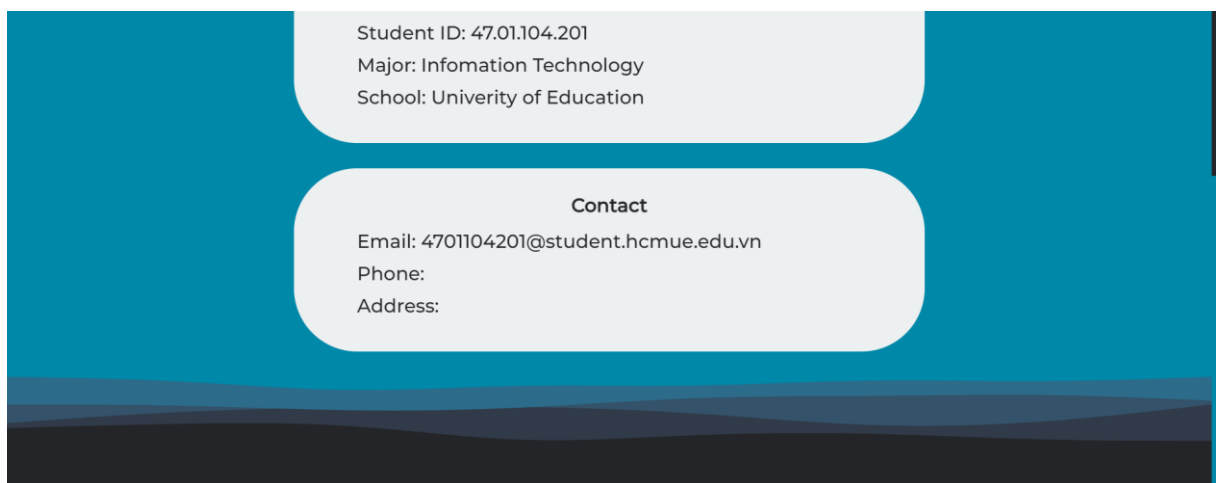
Hình 4.14. Giao diện trang cá nhân của My (tt)



Hình 4.15. Giao diện trang cá nhân của My (tt)



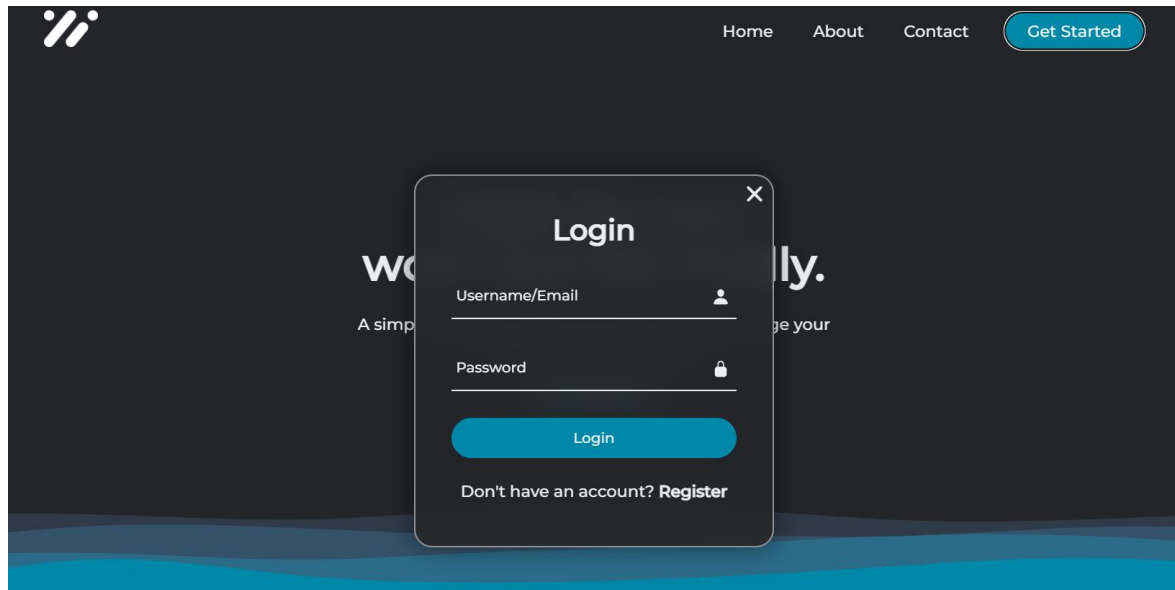
Hình 4.16. Giao diện trang cá nhân của Thịnh



Hình 4.17. Giao diện trang cá nhân của Thịnh (tt)

4.2. Đăng ký

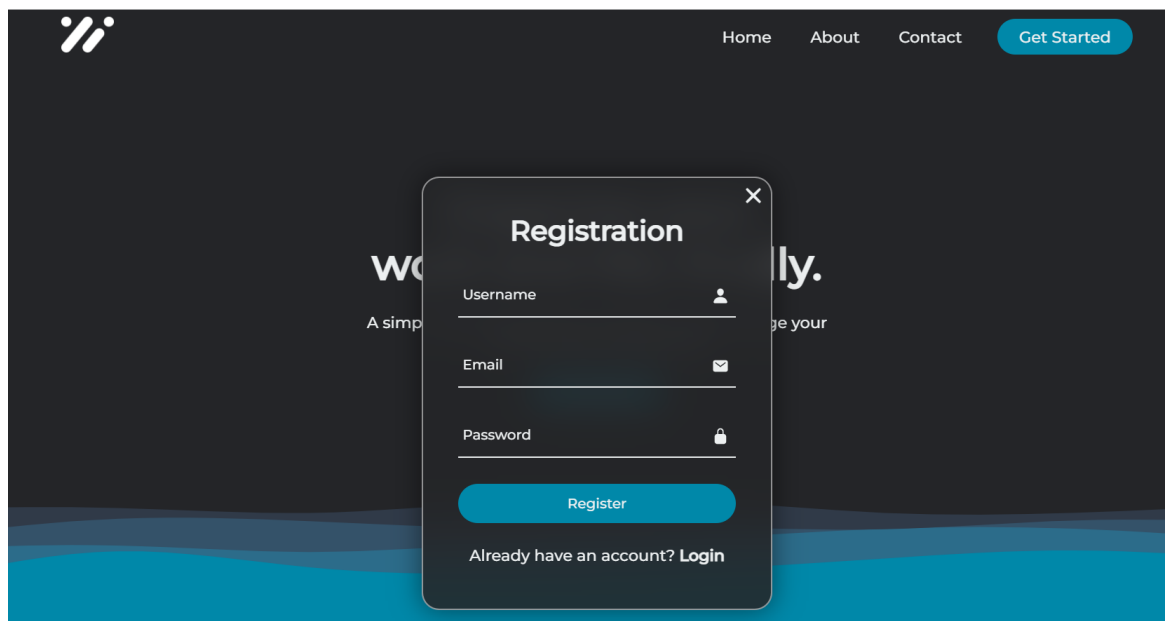
Tại trang giới thiệu người dùng muốn đăng nhập có thể nhấp vào **Get Started** ngay giữa hoặc trên góc phải của trang giới thiệu, trang đăng nhập sẽ xuất hiện ngay sau đó.



Hình 4.18. Giao diện đăng nhập

Nếu người dùng chưa có tài khoản thì có thể tiến hành đăng ký tài khoản. Tại trang login, hiển thị dòng **Don't have an account? Register**, người dùng nhấn vào chữ **Register** để đi tới trang đăng ký tài khoản.

Tiếp theo người dùng nhập đầy đủ thông tin trong form đăng ký (Username, Email, Password) và nhấn nút **Register** ngay bên dưới, hệ thống sẽ tiến hành lưu tài khoản vào trong database.

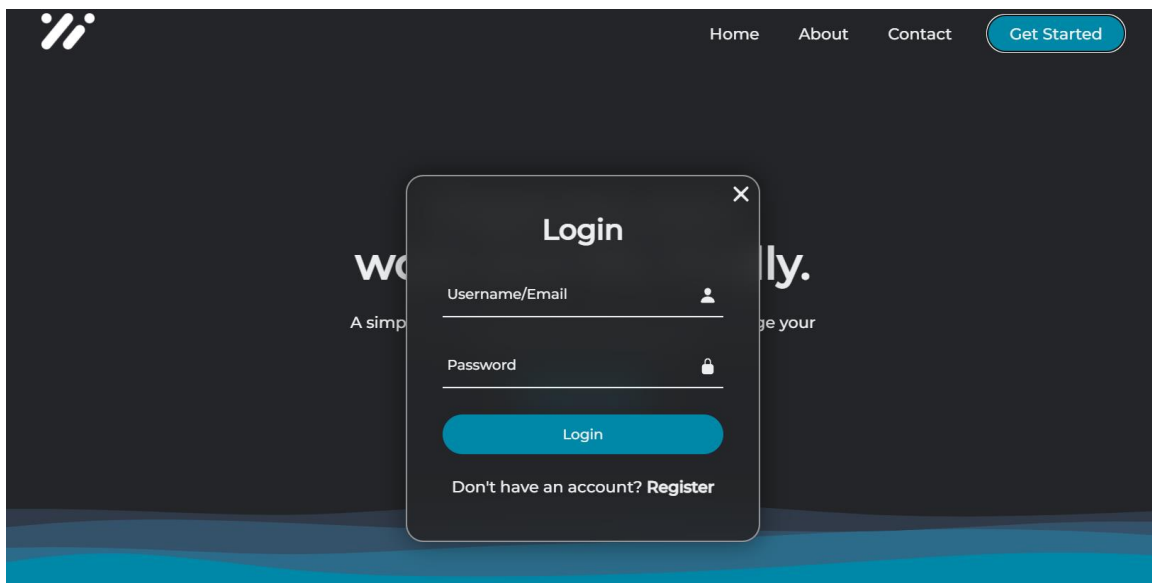


Hình 4.19. Giao diện đăng ký

Sau khi đăng ký thành công người dùng quay trở lại trang login để đăng nhập vào hệ thống.

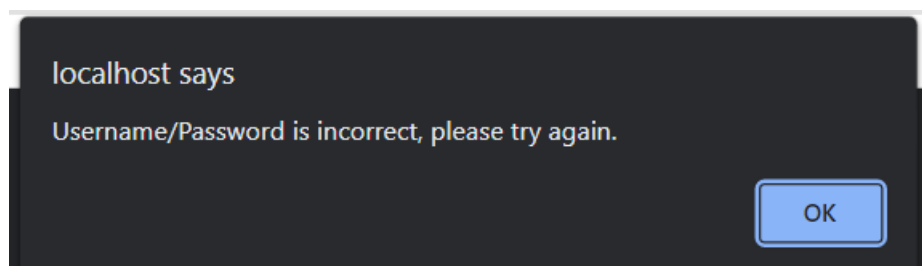
4.3. Đăng nhập

Tại trang giới thiệu người dùng muốn đăng nhập có thể nhấp vào **Get Started** ngay giữa hoặc trên góc phải của trang giới thiệu, trang đăng nhập sẽ xuất hiện ngay sau đó.



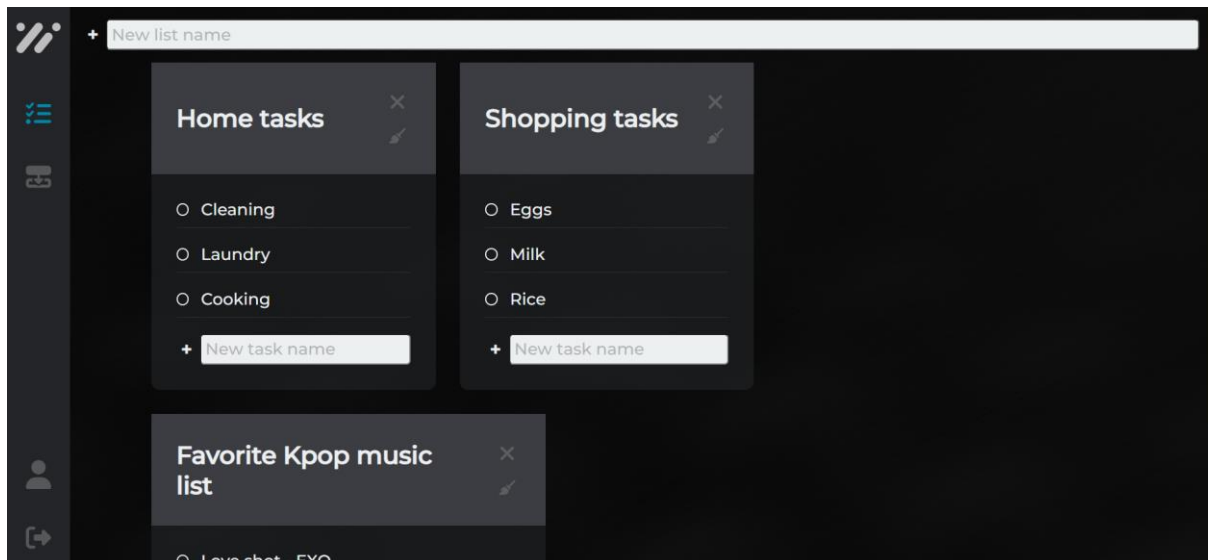
Hình 4.20. Giao diện đăng nhập

Nếu người dùng đăng nhập với tên đăng nhập và mật khẩu sai (không khớp với CSDL ban đầu) thì trang web sẽ hiển thị thông báo **“Username/Password is incorrect, please try again.”**.



Hình 4.21. Thông báo nhập sai Username/Password

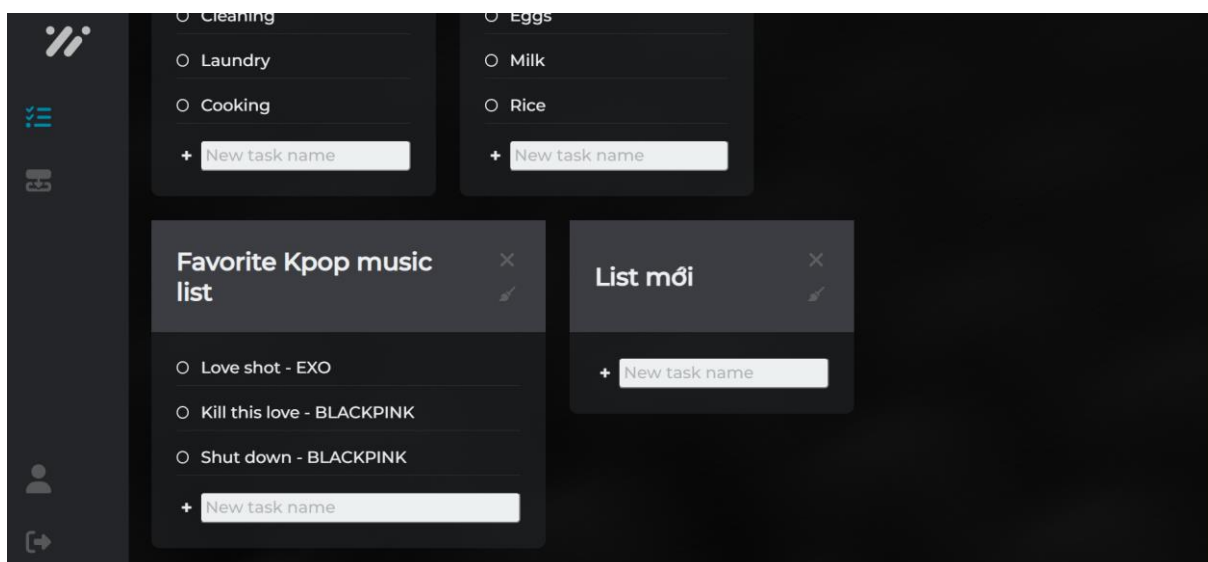
Nếu người dùng đăng nhập với tên đăng nhập và mật khẩu đúng (khớp với CSDL ban đầu) thì trang web sẽ tiến hành đăng nhập và đưa người dùng tới giao diện làm việc của web quản lý công việc (WorkFlow).



Hình 4.22. Giao diện trang WorkFlow (To-do List)

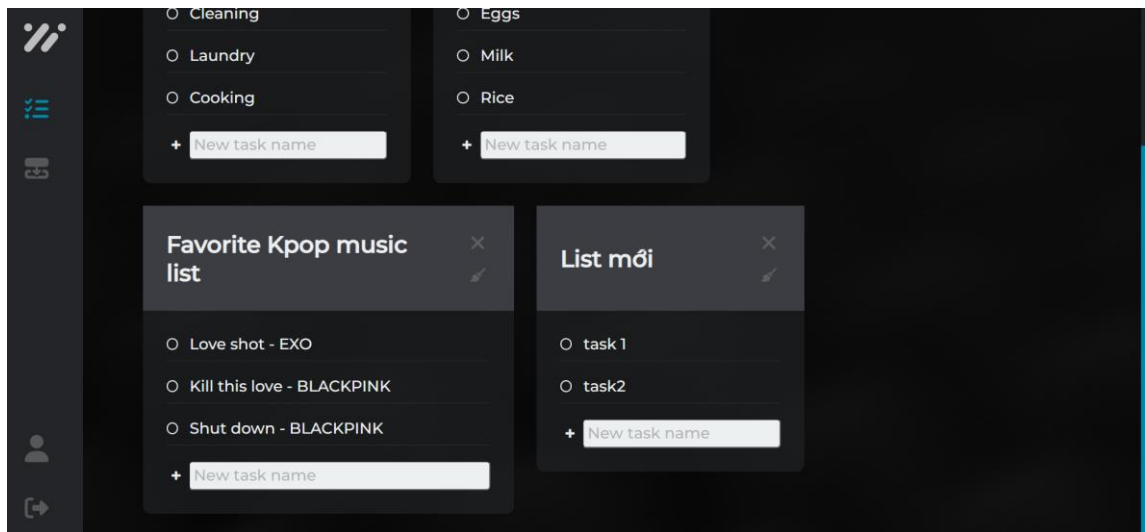
4.4. Todo-lists

Ở phần to-do list, tạo một list mới bằng cách nhập tên của list vào ô “**New list name**” và nhấn **Enter**. Sau đó list mới sẽ xuất hiện với tên mình vừa đặt.



Hình 4.23. Tạo New list

Để tạo các task cho list, ta nhập tên của task vào ô “**New task name**” và nhấn **Enter**.

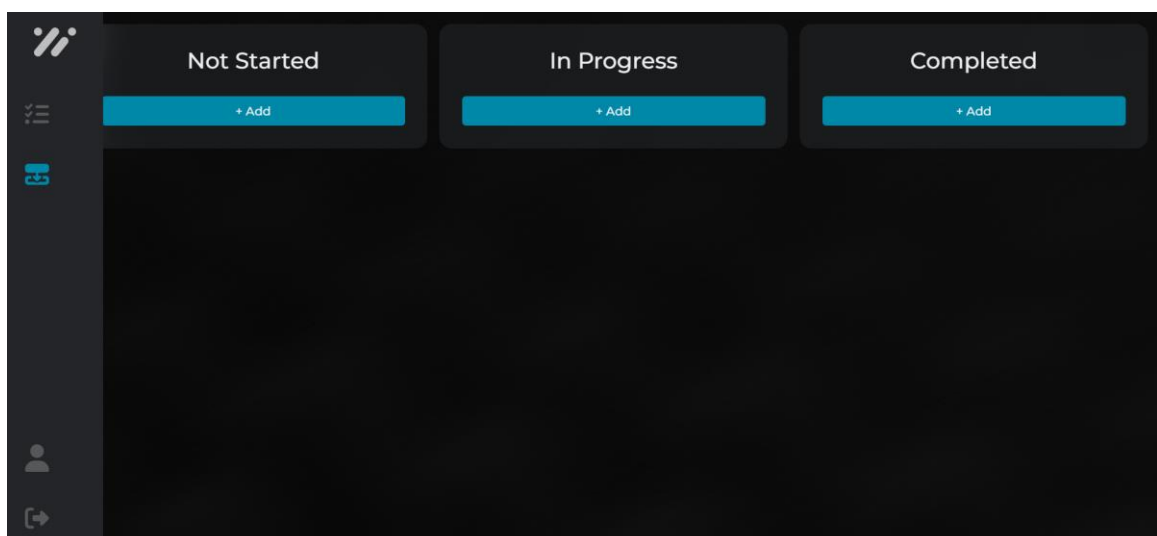


Hình 4.24. Tạo các task cho list

4.5. Task-board

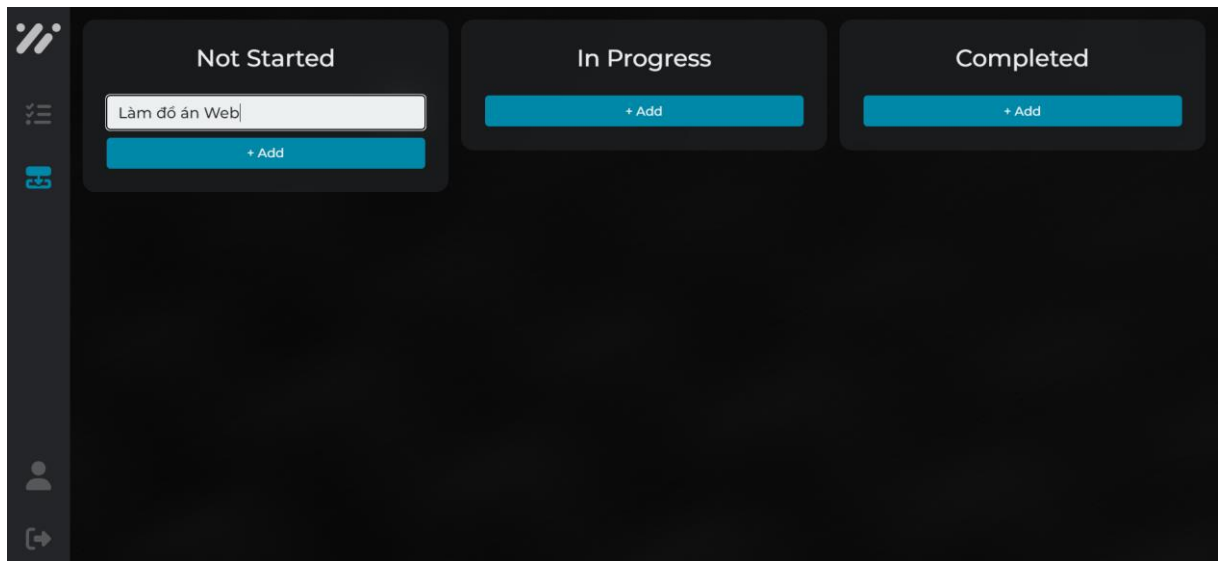
Task-board dùng để tổ chức các công việc để theo dõi tiến độ của đồ án, dự án, ... task đã được tạo được lưu lại sẽ không bị mất khi người dùng đăng xuất.

- Not Started: công việc dự định, chưa tiến hành thực hiện.
- In Progress: công việc đang trong quá trình thực hiện.
- Completed: công việc đã thực hiện xong.



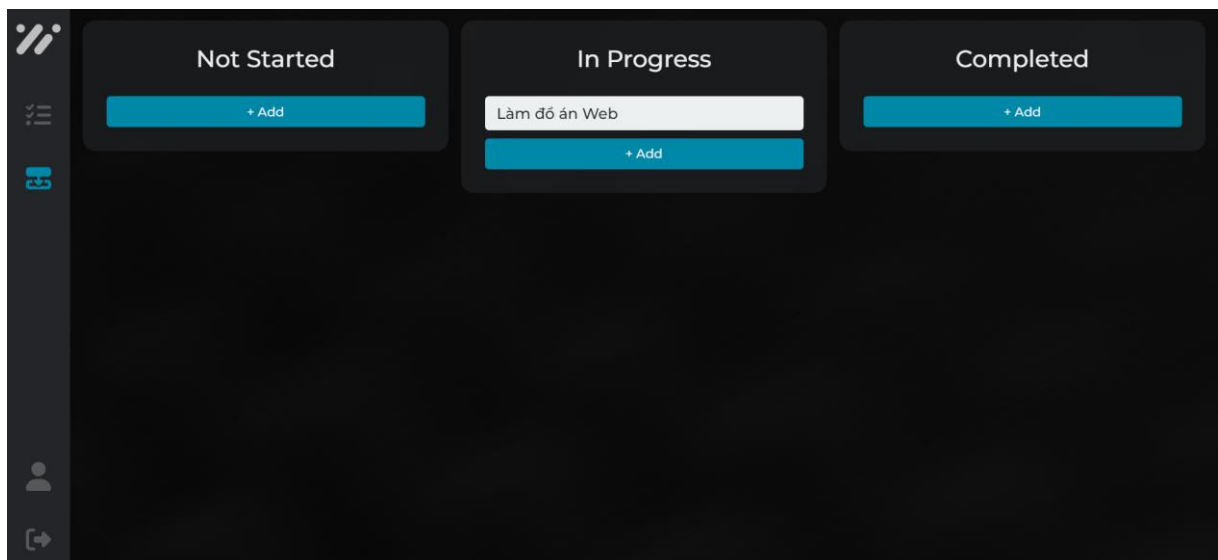
Hình 4.25. Giao diện task-board

Để tạo các nhiệm vụ mới, ta nhấn **Add**, sau đó nhập tên nhiệm vụ vào ô trống.



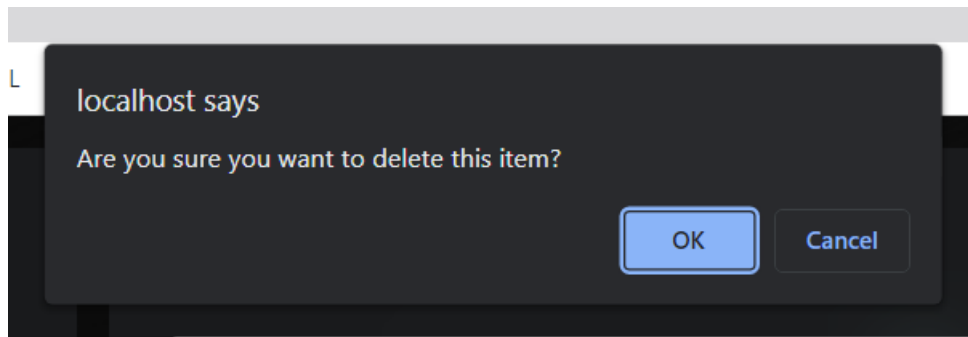
Hình 4.26. Tạo nhiệm vụ mới trong task-board

Chúng ta có thể kéo các thanh nhiệm vụ từ **Not Started** sang **In Progress** hoặc từ **In Progress** sang **Completed** tương ứng với tiến độ làm việc ở thực tế, giúp người dùng kiểm soát được tiến độ của dự án đi tới đâu, hoặc là số lượng công việc phải thực hiện khi làm dự án này.



Hình 4.27. Kéo nhiệm vụ vào ô tương ứng với tiến độ làm việc thực tế

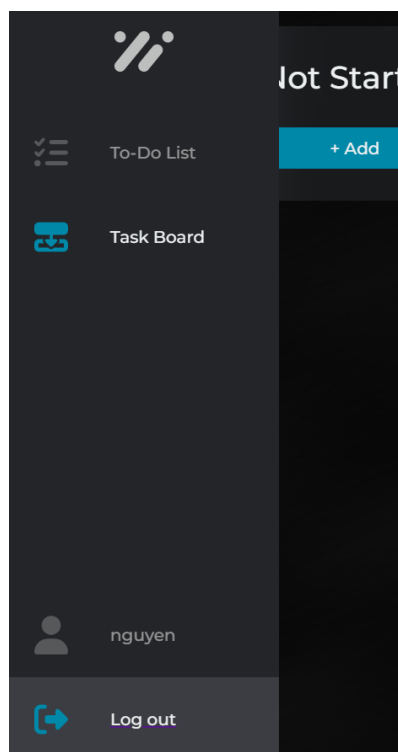
Khi muốn xóa đi một công việc tại task-board thì nhấp đôi vào thanh công việc muốn xóa hệ thống xuất hiện thông báo **“Are you sure you want to delete this item”**, chọn **OK** nếu xác định xóa công việc, **Cancel** nếu không cần xóa nữa.



Hình 4.28. Thông báo xóa nhiệm vụ

4.6. Đăng xuất

Người dùng muốn đăng xuất thì nhấn biểu tượng đăng xuất **“Log out”** ở góc dưới bên trái màn hình, trang web sẽ trở về giao diện trang giới thiệu.



Hình 4.29. Đăng xuất

CHƯƠNG 5. KẾT LUẬN

5.1. Những công việc đã làm được

- Ứng dụng đồ án quản lý công việc trên web đã đáp ứng được các yêu cầu đề ra ban đầu của mục tiêu đồ án.
- Hệ thống cho phép chạy trên nhiều nền tảng khác nhau bao gồm web server và web Browser và cả điện thoại.
- Về hệ thống quản trị cơ sở dữ liệu đồ án đã có thể để sử dụng sơ đồ thực thể kết hợp đạt dạng chuẩn 3.

5.2. Hướng phát triển của đồ án quản lý công việc trên web

- Đồ án quản lý công việc trên web cho phép mở rộng với các quy mô lớn hơn và khả năng trao đổi cơ sở dữ liệu giữa các hệ thống khác nhau.
- Người dùng có thể phát triển các ứng dụng cho phép download dữ liệu về máy tính và hệ thống quản lý.
- Hệ thống quản lý công việc trên web cho phép Quản lý hình ảnh về công việc của ứng dụng với thiết kế sinh động hơn.
- Cho phép cài đặt các ràng buộc toàn vẹn trên cơ sở dữ liệu của server.
- Xây dựng ứng dụng trên nhiều nền tảng hơn.
- Xây dựng giao diện tốt hơn, đẹp mắt hơn.

TÀI LIỆU THAM KHẢO

[1]. Đặc điểm của cơ sở dữ liệu (database) là gì?

<https://hoclaptrinh.vn/posts/dac-diem-cua-co-so-du-lieu-database-la-gi>

[2]. Cơ sở dữ liệu là gì? Các loại cơ sở dữ liệu hiện nay

<https://jobsgo.vn/blog/co-so-du-lieu-la-gi>

[3]. Ngôn ngữ lập trình PHP là gì? Tất tần tật những điều bạn cần biết về PHP

<https://topdev.vn/blog/ngon-ngu-lap-trinh-php-la-gi-tat-tan-tat-nhung-dieu-ban-can-biet-ve-php/>

[4]. Javascript là gì? Kiến thức tổng quan về Javascript từ A – Z

<https://fptcloud.com/javascript/>

[5]. CSS là gì?

<https://topdev.vn/blog/css-la-gi/>

[6]. HTML là gì? Ưu nhược điểm và vai trò của HTML trong lập trình web?

<https://bizflycloud.vn/tin-tuc/html-la-gi-20211119162558126.htm>

[7]. Tìm hiểu phần mềm XAMPP là gì? Các ưu nhược điểm của XAMPP

<https://adsplus.vn/blog/tim-hieu-phan-mem-xampp-la-gi-cac-uu-nhuoc-diem-cua-xampp/>