

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên : Trần Duy Hải

Giảng viên hướng dẫn: Th.sỹ Nguyễn Thị Xuân Hương

HẢI PHÒNG – 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

XÂY DỰNG WEBSITE
HELLEN TEA & COFFEE

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên : Trần Duy Hải

Giảng viên hướng dẫn: Ths. Nguyễn Thị Xuân Hương

HẢI PHÒNG – 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: **Trần Duy Hải**

Mã SV: **1612101006**

Lớp : **CT2001C**

Ngành : **Công nghệ thông tin**

Tên đề tài : **Xây dựng Website Hellen Tea & Coffee**

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

- Tìm hiểu các thông tin và hoạt động của quán Hellen Tea & Coffee.
- Phân tích thiết kế Website.
- Tìm hiểu ngôn ngữ lập trình PHP, hệ quản trị CSDL MySQL.
- Cài đặt Website.

2. Các tài liệu, số liệu cần thiết

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp

.....

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Nguyễn Thị Xuân Hương

Học hàm, học vị : Thạc sỹ

Cơ quan công tác : Trường Đại học Quản lý và Công nghệ Hải Phòng

Nội dung hướng dẫn: Xây dựng Website Hellen Tea & Coffee

Đề tài tốt nghiệp được giao ngày 12 tháng 10 năm 2020

Yêu cầu phải hoàn thành xong trước ngày 31 tháng 12 năm 2020

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày tháng năm 202

TRƯỜNG KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Nguyễn Thị Xuân Hương

Đơn vị công tác:

Họ và tên sinh viên: Trần Duy Hải, Ngành: Công nghệ Thông tin

Nội dung hướng dẫn: Tìm hiểu các thông tin và hoạt động của quán Hellen Tea & Coffee. Phân tích thiết kế Website. Tìm hiểu ngôn ngữ lập trình PHP, hệ quản trị CSDL MySQL, Cài đặt Website.

Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

- Sinh viên tích cực và nghiêm túc trong quá trình làm việc, tự học và tự nghiên cứu, tự học ngôn ngữ lập trình PHP để thực hiện đề án.
- Chấp hành tốt các yêu cầu và lịch trình làm việc do giáo viên đề ra.

Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T. T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...)

- Sinh viên Trần Duy Hải đã tìm hiểu hệ thống, phân tích thiết kế website và cài đặt được website cho quán Hellen Tea & coffee. Website đã được sử dụng cho quán và đáp ứng được các yêu cầu ban đầu đặt ra của người dùng và được đánh giá tốt.
- Tôi đề nghị cho phép sinh viên Trần Duy Hải được bảo vệ đề án tại hội đồng chấm đề án tốt nghiệp ngành Công nghệ Thông tin

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Đạt

☐

Không đạt

☐

Điểm:.....

Hải Phòng, ngày tháng năm 202

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: Ngành:

Đề tài tốt nghiệp:

.....

1. Phần nhận xét của giảng viên chấm phản biện

.....

.....

.....

.....

.....

2. Những mặt còn hạn chế

.....

.....

.....

.....

.....

.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm:.....

Hải Phòng, ngày tháng năm 202

Giảng viên chấm phản biện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong thời gian làm đồ án tốt nghiệp em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè. Em xin gửi lời cảm ơn chân thành đến Ths. Nguyễn Thị Xuân Hương, giảng viên Bộ môn Công Nghệ Thông Tin - Trường Đại học Quản lý và Công nghệ Hải Phòng người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình làm đồ án.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong trường Đại học Quản lý và Công nghệ Hải Phòng nói chung, các thầy cô trong Khoa Công Nghệ Thông Tin nói riêng đã dạy dỗ cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập để em có được ngày hôm nay.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 202

Sinh Viên

Trần Duy Hải

MỤC LỤC

GIỚI THIỆU	1
1. Mục tiêu đề tài.....	1
2. Giải pháp và cách thực hiện đề tài.....	1
3. Tìm hiểu về Hellen Tea & Coffee	1
4. Ý nghĩa	1
CHƯƠNG I: TỔNG QUAN	2
1.1 Tổng quan về World Wide Web.....	2
1.1.1 Khái niệm	2
1.1.2 Cách tạo ra website	2
1.1.3. Trình duyệt web (web Client hay web Browser).....	2
1.1.4. Webserver	3
1.2. Tổng quan về Html-Css.....	3
1.2.1. Giới thiệu về html.....	3
1.2.2. Giới thiệu về Css.....	4
1.3. Ngôn ngữ PHP.....	5
1.3.1 Định nghĩa PHP	5
1.3.2. Lý do nên dùng PHP	6
1.3.3. Hoạt động của PHP	6
1.3.4. Tổng quan về PHP	7
1.3.5. Hàm trong PHP	8
1.3.6. Biểu mẫu PHP	9
1.3.7. PHP OOP(Object-Oriented Programming.)	9
1.3.8. Session và Cookie	9
1.4. My SQL.....	10
1.5.Giới thiệu về Bootstrap.....	10
1.5.1. Bootstrap là gì?.....	10
1.5.2. Lịch sử của Bootstrap	11

1.5.3. Tại sao nên sử dụng Bootstrap?	11
1.5.4. Cấu trúc và tính năng của Bootstrap là gì?	12
1.6. Giới thiệu về jQuery	12
1.6.1. Định nghĩa về jQuery	12
1.6.2. Tại sao nên chọn jQuery	13
1.6.3. Cài đặt tích hợp jQuery vào website	13
1.6.4. Cú pháp jQuery	14
1.7 Tìm hiểu về mô hình MVC.....	14
1.7.1. MVC là gì?	14
1.7.2. Các thành phần trong MVC	15
1.7.3. MVC làm việc như thế nào?.....	16
1.7.4. Ưu điểm và nhược điểm của MVC	16
CHƯƠNG II: THIẾT KẾT BACK-END, FRONT-END	17
1.Phân tích thiết kế hệ thống.....	17
1.1. Thiết kế cơ sở dữ liệu	20
1.2. Lược đồ quan hệ.....	26
1.3. Mô hình thực thể E-R	27
1.4. Mô hình liên kết dữ liệu.....	28
1.5. Sơ đồ phân rã chức năng	29
2. Phân tích cấu trúc thư mục.....	30
2.1. Controller	30
2.2. Model	31
2.3. View	32
3. Xây dựng Back-end	33
3.1. Các chức năng quản lý đồ uống	34
3.2 Các chức năng quản lý Banner Slider.....	44
3.3. Các chức năng quản lý bài viết	50
3.4. Các chức năng quản lý tài khoản	58
3.5. Các chức năng đăng nhập và phân quyền	65

4. Xây dựng Front-end	71
4.1. Trang chủ	73
4.2. Thanh điều hướng	77
4.3. Trang danh sách đồ uống	78
4.4. Trang danh sách bài viết	82
4.5. Banner Slider	86
4.6. Footer	87
CHƯƠNG III: DEMO WEBSITE	89
1. Demo Back-end	89
2. Demo Front-end	94
2.1. Demo trang chủ	94
2.2. Demo trang bài viết.....	95
2.3. Demo trang đồ uống.....	96
2.4. Demo trang giới thiệu	98
2.5. Demo trang liên hệ	98
KẾT LUẬN	99
TÀI LIỆU THAM KHẢO.....	100

LỜI MỞ ĐẦU

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức, cũng như của các công ty, doanh nghiệp. Điều này đóng vai trò hết sức quan trọng và là một trong những nền tảng để tạo ra những bước đột phá mạnh mẽ.

Cùng với sự phát triển không ngừng về kỹ thuật máy tính và mạng Internet, công nghệ thông tin đã chứng minh được rằng đây là một trong những ngành mũi nhọn, lần lượt chinh phục hết đỉnh cao này đến đỉnh cao khác. Trên thực tế, mạng Internet là một trong những sản phẩm có giá trị và mức độ ảnh hưởng to lớn và trở thành một công cụ không thể thiếu, là nền tảng chính để giúp truyền tải, trao đổi thông tin trên toàn cầu.

Ngày nay, mọi việc liên quan đến thông tin trở nên thật dễ dàng cho người dùng, chỉ cần có một máy tính kết nối Internet và một dòng dữ liệu truy tìm thì gần như lập tức, cả thế giới về vấn đề mà bạn đang quan tâm sẽ hiện ra, có đầy đủ thông tin, hình ảnh và thậm chí đôi lúc có cả những âm thanh nếu bạn cần, v.v.

Thông qua Internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này, đã thúc đẩy sự khai sinh và phát triển của thương mại điện tử và chính phủ điện tử trên khắp thế giới, làm biến đổi đáng kể về văn hóa, nâng cao chất lượng cuộc sống con người.

Trong hoạt động sản xuất, kinh doanh, thương mại điện tử đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của doanh nghiệp. Đối với một cửa hàng hay shop, việc quảng bá và giới thiệu đến khách hàng các sản phẩm mới đáp ứng được nhu cầu của khách hàng là cực kỳ quan trọng và cần thiết. Một trong những công cụ hữu ích là xây dựng được một Website cho cửa hàng của mình để giới thiệu và quảng bá tất cả các sản phẩm, dịch vụ đến người dùng.

Chính vì lý do này, em đã chọn thực hiện đề án “**Xây dựng Website Hellen Tea & Coffee**” làm đề án tốt nghiệp của mình để có thể xây dựng một website hữu ích cho người dùng.

Với sự hướng dẫn tận tình của cô giáo *Nguyễn Thị Xuân Hương* em đã hoàn thành đề án này. Tuy đã cố gắng hết sức tìm hiểu, phân tích thiết kế và cài đặt hệ thống nhưng chắc rằng sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu của Thầy Cô. Em xin chân thành cảm ơn

GIỚI THIỆU

1. Mục tiêu đề tài

Tìm hiểu, nghiên cứu và xây dựng Website Hellen Tea & Coffee

Xây dựng một website động với nội dung linh hoạt, cấu trúc rõ ràng, bắt mắt để phù hợp với mục đích quảng bá, bán hàng.

2. Giải pháp và cách thực hiện đề tài

Tìm hiểu về các website đồ uống phổ biến, tham khảo về bố cục của website. Đến quán Hellen Tea để tìm hiểu các thông tin và hoạt động của quán, thu thập hình ảnh, dữ liệu. Sau đó phân tích thiết kế hệ thống thông tin, xây dựng nội dung quản trị Back-end và giao diện Front-end website.

Xây dựng một website thì việc lựa chọn ngôn ngữ và ứng dụng khi làm việc là điều rất quan trọng. Đề tài sử dụng ngôn ngữ PHP làm ngôn ngữ lập trình để xây dựng website và một số thư viện hỗ trợ trong việc xây dựng website.

3. Tìm hiểu về Hellen Tea & Coffee

Quán Hellen Tea & Coffee là nơi kinh doanh đồ uống, được khai trương vào tháng 8 năm 2020. Quán thuộc quyền sở hữu của gia đình, có quy mô khá nhỏ (sức chứa khoảng 30 người). Tại đó bán đa dạng các loại đồ uống như: trà sữa, coffee, sinh tố, v.v. Hellen Tea & Coffee còn nhận tổ chức các sự kiện có quy mô vừa và nhỏ ví dụ như: tiệc, sinh nhật, v.v. Ngoài việc bán hàng trực tiếp tại thì quán còn bán online và giao hàng tận nơi sử dụng dịch vụ trên Foody.vn.

Địa chỉ: 147B Trung Lực, P. Đằng Lâm, Quận Hải An, Hải Phòng

4. Ý nghĩa

Lợi ích của việc xây dựng website nói chung và website Hellen Tea & Coffee nói riêng là điều rất cần thiết và quan trọng. Giúp ích trong việc truyền đạt thông tin tới người mua hàng, quảng bá hình ảnh của quán. Giúp các thực khách cập nhật được tin tức thú vị xoay quanh chủ đề đồ uống, thông báo khuyến mãi mới và quan trọng là giúp ích cho quá trình mua sản phẩm của quán thông qua internet.

CHƯƠNG I: TỔNG QUAN

1.1 Tổng quan về World Wide Web

1.1.1 Khái niệm

World Wide Web (WWW) hay còn gọi là web là một dịch vụ phổ biến nhất hiện nay trên Internet, 85% các giao dịch trên Internet ước lượng thuộc về WWW. Ngày nay số website trên thế giới đã đạt tới con số khổng lồ. WWW cho phép truy xuất thông tin văn bản, hình ảnh, âm thanh, video trên toàn thế giới. Thông qua website, các quý công ty có thể giảm thiểu tối đa chi phí in ấn và phân phát tài liệu cho khách hàng ở nhiều nơi.

1.1.2 Cách tạo ra website

Có nhiều cách để tạo website, có thể tạo website trên bất kì chương trình xử lí văn bản nào:

Tạo web bằng cách viết mã nguồn bởi một trình soạn thảo văn bản như:

- Notepad, WordPad, v.v. là những chương trình soạn thảo văn bản có sẵn trong Window.
- Thiết kế bằng cách dùng web Wizard và công cụ của Word 97, Word 2000.
- Thiết kế web bằng các phần mềm chuyên nghiệp: PHPStorm, Dreamweaver, Nescape Editor,.. sẽ giúp thiết kế website dễ dàng hơn, nhanh chóng hơn, phần lớn mã lệnh HTML sẽ có sẵn trong phân code.
- Để xây dựng một ứng dụng web hoàn chỉnh và có tính thương mại, cần kết hợp cả Client Script (kịch bản trình khách) và Server Script (kịch bản trên trình chủ) với một loại cơ sở dữ liệu nào đó, chẳng hạn như: MSAccess, SQL Server, MySQL, Oracle, v.v.
- Khi muốn triển khai ứng dụng web trên mạng, ngoài các điều kiện về cấu hình phần cứng, cần có trình chủ web thường gọi là web Server.

1.1.3. Trình duyệt web (web Client hay web Browser)

Trình duyệt Web là công cụ truy xuất dữ liệu trên mạng, là phần mềm giao diện trực tiếp với người sử dụng. Nhiệm vụ của Web Browser là nhận các yêu cầu của người dùng, gửi các yêu cầu đó qua mạng tới các Web Server và nhận các dữ liệu cần thiết từ Server để hiển thị lên màn hình. Để sử dụng dịch vụ WWW, Client cần có một chương trình duyệt Web, kết nối vào Internet thông qua một ISP. Các trình duyệt thông dụng hiện nay là: Microsoft Internet Explorer, Google Chrome, Mozilla FireFox, v.v.

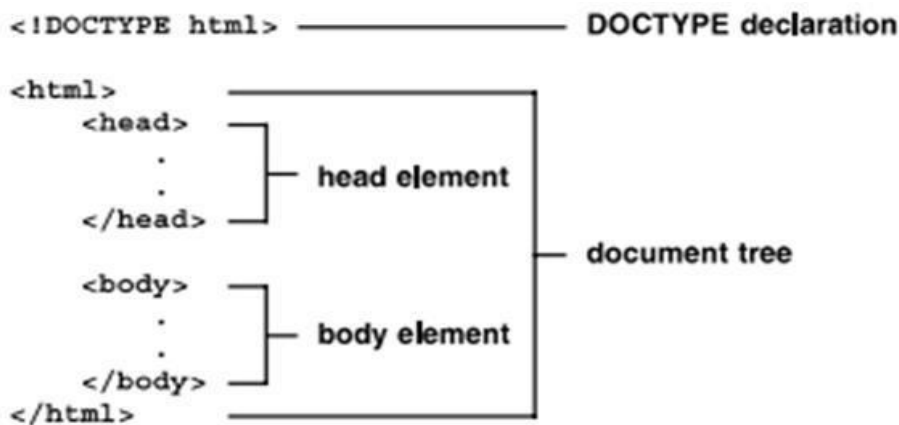
1.1.4. Webserver

"Web server" có thể là phần cứng hoặc phần mềm, hoặc cả hai.

- Ở khía cạnh phần cứng, một web server là một máy tính lưu trữ các file thành phần của một website (ví dụ: các tài liệu HTML, các file ảnh, CSS và các file JavaScript) và có thể phân phát chúng tới thiết bị của người dùng cuối (end-user). Nó kết nối tới mạng Internet và có thể truy cập tới thông qua một tên miền giống như mozilla.org.
- Ở khía cạnh phần mềm, một web server bao gồm một số phần để điều khiển cách người sử dụng web truy cập tới các file được lưu trữ trên một HTTP server(máy chủ HTTP). Một HTTP server là một phần mềm hiểu được các URL (các địa chỉ web) và HTTP (giao thức trình duyệt của bạn sử dụng để xem các website).
- IIS (Internet Information Service): Hỗ trợ ASP, mở rộng hỗ trợ PHP.
- Apache: Hỗ trợ PHP.
- Tomcat: Hỗ trợ JSP (Java Servlet Page).

1.2. Tổng quan về Html-Css

1.2.1. Giới thiệu về html



Hình 1.1: Cấu trúc website

- Thẻ `<!DOCTYPE html>` định nghĩa trang html, kiểu khai báo của html 5.
- Thẻ tạo tiêu đề trang trên thanh tiêu đề, đây là thẻ bắt buộc. Thẻ title cho phép trình bày chuỗi trên thanh tựa đề của website mỗi khi website đó được duyệt trên trình duyệt web.

- Thẻ <body> ... </body> tất cả các thông tin khai báo trong thẻ <body> đều có thể xuất hiện trên website. Những thông tin này có thể nhìn thấy trên website.
- Thẻ <div>.....</div> định nghĩa một nội dung.
- Thẻ <p> ... </p> tạo một đoạn mới..
- Thẻ ... thay đổi phông chữ, kích cỡ và màu kí tự.
- Thẻ <table> ... </table> đây là thẻ định dạng bảng trên website. Sau khi khai báo thẻ này, phải khai báo các thẻ hàng <tr> và thẻ cột <td> cùng với các thuộc tính của nó.
- Thẻ cho phép chèn hình ảnh vào website. Thẻ này thuộc loại thẻ không có thẻ đóng.
- Thẻ <a> ... là loại thẻ dùng để liên kết giữa các website hoặc liên kết đến địa chỉ Internet, Mail hay Intranet (URL) và địa chỉ trong tập tin trong mạng cục bộ (UNC).
- Thẻ <input /> cho phép người dùng nhập dữ liệu hay chỉ thị thực thi một hành động nào đó, thẻ Input bao gồm các loại thẻ như: text, password, submit, button, reset, checkbox, radio, hidden, image.
- Thẻ < textarea>.... < \textarea> cho phép người dùng nhập liệu với rất nhiều dòng. Với thẻ này không thể giới hạn chiều dài lớn nhất trên website.
- Thẻ <select> ... </select> cho phép người dùng chọn phần tử trong tập phương thức đã được định nghĩa trước. Nếu thẻ <select> cho phép người dùng chọn một phần tử trong danh sách phần tử thì thẻ <select> sẽ giống như combobox. Nếu thẻ <select> cho phép người dùng chọn nhiều phần tử cùng một lần trong danh sách phần tử, thẻ <select> đó là dạng listbox.
- Thẻ <form> ... </form> khi muốn submit dữ liệu người dùng nhập từ website phía Client lên phía Server, có hai cách để làm điều này ứng với hai phương thức POST và GET trong thẻ form. Trong một website có thể có nhiều thẻ <form> khác nhau, nhưng các thẻ <form> này không được lồng nhau, mỗi thẻ form sẽ được khai báo hành động (action) chỉ đến một trang khác.

1.2.2. Giới thiệu về Css

Css là ngôn ngữ tạo phong cách cho website – Cascading Style Sheet language.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996.

Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều website

cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của website với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ, v.v.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

Các stylesheet ngoài được lưu trữ dưới dạng các tập tin .CSS.

Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn responsive. Bootstrap cho phép quá trình dùng để phát triển website chuẩn responsive. Bootstrap cho phép quá trình thiết kế website diễn ra nhanh chóng và dễ dàng hơn.

1.3. Ngôn ngữ PHP

1.3.1 Định nghĩa PHP

PHP là chữ viết tắt của “Personal Home Page do” Rasmus Lerdorf tạo ra năm 1994. Vì tính hữu dụng của nó và khả năng phát triển, PHP bắt đầu được sử dụng trong môi trường chuyên nghiệp và nó trở thành “PHP:Hypertext Preprocessor”. Thực chất PHP là ngôn ngữ kịch bản nhúng trong HTML, nói một cách đơn giản đó là một trang HTML có nhúng mã PHP, PHP có thể được đặt rải rác trong HTML.

PHP là một ngôn ngữ lập trình được kết nối chặt chẽ với máy chủ, là một công nghệ phía máy chủ (Server-Side) và không phụ thuộc vào môi trường (crossplatform). Đây là hai yếu tố rất quan trọng, thứ nhất khi nói công nghệ phía máy chủ tức là nói đến mọi thứ trong PHP đều xảy ra trên máy chủ, thứ hai, chính vì tính chất không phụ thuộc môi trường cho phép PHP chạy trên hầu hết trên các hệ điều hành như Windows, Unix và nhiều biến thể của nó, v.v. Đặc biệt các mã kịch bản PHP viết trên máy chủ này sẽ làm việc bình thường trên máy chủ khác mà không cần phải chỉnh sửa hoặc chỉnh sửa rất ít.

PHP chạy trên môi trường Webserver và lưu trữ dữ liệu thông qua hệ quản trị cơ sở dữ liệu nên PHP thường đi kèm với Apache, MySQL, v.v.

Apache là một phần mềm web server có nhiệm vụ tiếp nhận request từ trình duyệt người dùng sau đó chuyển giao cho PHP xử lý và gửi trả lại cho trình duyệt.

MySQL cũng tương tự như các hệ quản trị cơ sở dữ liệu khác (Postgress, Oracle, SQL server, v.v.) đóng vai trò là nơi lưu trữ và truy vấn dữ liệu.

1.3.2. Lý do nên dùng PHP

PHP được sử dụng làm web động vì nó nhanh, dễ dàng, tốt hơn so với các giải pháp khác.

PHP có khả năng thực hiện và tích hợp chặt chẽ với hầu hết các cơ sở dữ liệu có sẵn, tính linh động, bền vững và khả năng phát triển không giới hạn.

Đặc biệt PHP là mã nguồn mở do đó tất cả các đặc tính trên đều miễn phí, và chính vì mã nguồn mở sẵn có nên cộng đồng các nhà phát triển web luôn có ý thức cải tiến nó, nâng cao để khắc phục các lỗi trong các chương trình này.

PHP vừa dễ với người mới sử dụng vừa có thể đáp ứng mọi yêu cầu của các lập trình viên chuyên nghiệp, mọi ý tưởng của các PHP có thể đáp ứng một cách xuất sắc. PHP đã có mặt trên 13 triệu website.

1.3.3. Hoạt động của PHP

Vì PHP là ngôn ngữ của máy chủ nên mã lệnh của PHP sẽ tập trung trên máy chủ để phục vụ các website theo yêu cầu của người dùng thông qua trình duyệt.

- Sơ đồ hoạt động



Hình 1.2: Sơ đồ hoạt động của PHP

Khi người dùng truy cập website viết bằng PHP, máy chủ đọc mã lệnh PHP và xử lý chúng theo các hướng dẫn được mã hóa. Mã lệnh PHP yêu cầu máy chủ gửi một dữ liệu thích hợp (mã lệnh HTML) đến trình duyệt web. Trình duyệt xem nó như là một trang HTML tiêu chuẩn. Như ta đã nói, PHP cũng chính là một trang HTML nhưng có nhúng mã PHP và có phần mở rộng là HTML. Phần mở của PHP được đặt trong thẻ mở. Khi trình duyệt truy cập vào một trang PHP, Server sẽ đọc nội dung file PHP lên và lọc ra các đoạn mã PHP và thực thi các đoạn mã đó, lấy kết quả nhận được của đoạn mã PHP thay thế vào chỗ ban đầu của chúng trong file PHP, cuối cùng Server trả về kết quả cuối cùng là một trang nội dung HTML về cho trình duyệt.

1.3.4. Tổng quan về PHP

Cấu trúc cơ bản: PHP cũng có thể bắt đầu và kết thúc giống với HTML

Cú pháp chính

`<?php Mã lệnh php ?>`

Trong PHP để kết thúc 1 dòng lệnh chúng ta sử dụng dấu ";". Để chú thích một đoạn dữ liệu nào đó trong PHP ta sử dụng dấu "//" cho từng dòng hoặc dùng cặp thẻ "/* */" cho từng cụm mã lệnh.

Ví dụ: `<?php echo "Hello world!"; ?>`

Xuất giá trị ra trình duyệt chúng ta có những dòng cú pháp sau:

Echo "Thông tin";

Printf "Thông tin"; Thông tin bao gồm biến, chuỗi, hoặc lệnh html....

PHP có các kiểu dữ liệu cơ bản sau:

- Số nguyên, số thực, chuỗi, Boolean, mảng, đối tượng, resource, NULL/

Một số thành phần chính trong PHP:

- Biến: Một biến bắt đầu bằng dấu \$, theo sau là tên của biến.
- Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới.
- Một tên biến không thể bắt đầu bằng một số.
- Tên biến chỉ có thể chứa các ký tự chữ và số dưới (Az, 0-9 và _).
- Tên biến là phân biệt chữ hoa chữ thường.

Chuỗi: là một nhóm các ký tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy, ví dụ: 'Hello'.

Hằng: Một hằng số là một định danh (tên) cho một giá trị đơn giản. Giá trị không thể thay đổi trong tập lệnh. Để tạo một hằng số, sử dụng define() hàm. Các hằng số được tự động toàn cầu và có thể được sử dụng trên toàn bộ tập lệnh.

1.3.5. Hàm trong PHP

Sức mạnh thực sự của PHP đến từ các Hàm

PHP có hơn 1000 hàm dựng sẵn và ngoài ra, bạn có thể tạo các hàm tùy chỉnh.

Để giảm thời gian lặp lại 1 thao tác code nhiều lần, PHP hỗ trợ người lập trình việc tự định nghĩa cho mình những hàm có khả năng lặp lại nhiều lần trong website. Việc này cũng giúp cho người lập trình kiểm soát mã nguồn một cách mạch lạc, đồng thời có thể tùy biến ở mọi trang mà không cần phải khởi tạo hay viết lại mã lệnh như HTML thuần.

Một khai báo hàm do người dùng định nghĩa bắt đầu bằng từ function, tên hàm phải bắt đầu bằng chữ cái hoặc dấu ngoặc dưới. Tên hàm không phân biệt chữ hoa, chữ thường.

Tự định nghĩa hàm

```
Function functionName(){  
    // Lệnh thực thi;  
}
```

Tự định nghĩa hàm có tham số

```
Function functionName($giatri1, $giatri2){  
    // Lệnh thực thi;  
}
```

Tự định nghĩa hàm có giá trị trả về

```
Function functionName(){  
    // Lệnh thực thi;  
    Return giatrij;
```

Hàm có đối số

```
Function functionName($giatri = 1){  
    // Lệnh thực thi;  
}
```

1.3.6. Biểu mẫu PHP

Có 2 phương thức được sử dụng trong lập trình PHP là GET và POST.

- Phương thức GET: cũng được dùng để lấy dữ liệu từ form nhập liệu. Tuy nhiên nhiệm vụ chính của nó vẫn là lấy nội dung trang dữ liệu từ web server.
- Phương thức POST: phương thức này được sử dụng để lấy dữ liệu từ form nhập liệu và chuyển chúng lên trình chủ webserver.

1.3.7. PHP OOP(Object-Oriented Programming.)

OOP là viết tắt của lập trình hướng đối tượng, là việc tạo các đối tượng chứa cả hàm và dữ liệu.

Lập trình hướng đối tượng có một số lợi thế so với lập trình thủ tục:

- OOP nhanh hơn và dễ thực hiện hơn.
- OOP cung cấp một cấu trúc rõ ràng cho các chương trình.
- OOP giúp giữ mã PHP DRY "Đừng lặp lại chính mình" và làm cho mã dễ dàng hơn để duy trì, sửa đổi và gỡ lỗi.
- OOP cho phép tạo các ứng dụng có thể tái sử dụng đầy đủ với ít mã hơn và thời gian phát triển ngắn hơn.

1.3.8. Session và Cookie

Cookie và Session là hai phương pháp sử dụng để quản lý các phiên làm việc giữa người sử dụng và hệ thống.

Cookie và Session là hai phương pháp sử dụng để quản lý các phiên làm việc giữa người sử dụng và hệ thống.

Session dùng để lưu giữ liệu trên server, session dùng để lưu trữ thông tin người dùng, hoặc lưu trữ tùy chọn cấu hình hệ thống cho người dùng. Tất cả session đc lưu trữ trong biến toàn cục \$_SESSION.

Cookie dùng để lưu trữ các tùy chọn riêng của website với từng user, nó là file nhỏ được chỉ định lưu trên máy tính client và có thể truy xuất được, cần trình duyệt hỗ trợ chức năng này. Cookie không bị mất khi bị đóng ứng dụng lại, chỉ mất khi hết hạn thời gian thiết lập. Tất cả cookie được lưu trữ trong biến toàn cục \$_COOKIE.

1.4. My SQL

- Giới thiệu về My SQL:

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là cơ sở dữ liệu có trình giao diện trên Windows hay Linux, cho phép người sử dụng có thể thao tác các hành động liên quan đến nó. Việc tìm hiểu từng công nghệ trước khi bắt tay vào việc viết mã kịch bản PHP, việc tích hợp hai công nghệ PHP và MySQL là một công việc cần thiết và rất quan trọng.

- Mục đích sử dụng cơ sở dữ liệu:

Lưu trữ: Lưu trữ trên đĩa và có thể chuyển đổi dữ liệu từ cơ sở dữ liệu này sang cơ sở dữ liệu khác, nếu sử dụng cho quy mô nhỏ, có thể chọn cơ sở dữ liệu nhỏ như: Microsoft Excel, Microsoft Access, MySQL, Microsoft Visual FoxPro,, v.v. Nếu ứng dụng có quy mô lớn, có thể chọn cơ sở dữ liệu có quy mô lớn như: Oracle, SQL Server,, v.v.

Truy cập: Truy cập dữ liệu phụ thuộc vào mục đích và yêu cầu của người sử dụng, ở mức độ mang tính cục bộ, truy cập cơ sở dữ liệu ngay trong cơ sở dữ liệu với nhau, nhằm trao đổi hay xử lý dữ liệu ngay bên trong chính nó, nhưng do mục đích và yêu cầu người dùng vượt ra ngoài cơ sở dữ liệu, nên cần có các phương thức truy cập dữ liệu giữa các cơ sở dữ liệu với nhau như: Microsoft Access với SQL Server, hay SQL Server và cơ sở dữ liệu Oracle, v.v.

Tổ chức: Tổ chức cơ sở dữ liệu phụ thuộc vào mô hình cơ sở dữ liệu, phân tích và thiết kế cơ sở dữ liệu tức là tổ chức cơ sở dữ liệu phụ thuộc vào đặc điểm riêng của từng ứng dụng. Tuy nhiên khi tổ chức cơ sở dữ liệu cần phải tuân theo một số tiêu chuẩn của hệ thống cơ sở dữ liệu nhằm tăng tính tối ưu khi truy cập và xử lý.

Xử lý: Tùy vào nhu cầu tính toán và truy vấn cơ sở dữ liệu với các mục đích khác nhau, cần phải sử dụng các phát biểu truy vấn cùng các phép toán, phát biểu của cơ sở dữ liệu để xuất ra kết quả như yêu cầu. Để thao tác hay xử lý dữ liệu bên trong chính cơ sở dữ liệu ta sử dụng các ngôn ngữ lập trình như: PHP, C++, Java, Visual Basic, v.v.

1.5. Giới thiệu về Bootstrap

1.5.1. Bootstrap là gì?

- Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn responsive.

- Bootstrap cho phép quá trình thiết kế website diễn ra nhanh chóng và dễ dàng hơn dựa trên những thành tố cơ bản sẵn có như typography, forms, buttons, tables, grids, navigation, image carousels...

- Bootstrap là một bộ sưu tập miễn phí của các mã nguồn mở và công cụ dùng để tạo ra một mẫu website hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng..., các designer có thể sáng tạo nhiều sản phẩm mới mẻ nhưng vẫn tiết kiệm thời gian khi làm việc với framework này trong quá trình thiết kế giao diện website.

1.5.2. Lịch sử của Bootstrap

- Bootstrap là sản phẩm của Mark Otto và Jacob Thornton tại Twitter. Nó được xuất bản như là một mã nguồn mở vào ngày 19/8/2011 trên GitHub. Tên gọi ban đầu là Twitter Blueprint.

- Đến ngày 31/1/2012, Bootstrap phiên bản 2 đã được phát hành. Bootstrap 2 được bổ sung bố cục lưới 12 cột với thiết kế tùy chỉnh đáp ứng cho nhiều màn hình kích thước. Tiếp nối sự thành công của phiên bản 2, Bootstrap 3 ra đời vào ngày 19/8/2013 với giao diện tương thích với smartphone.

Chỉ 3 năm sau ngày ra mắt, Bootstrap đã trở thành No.1 project trên GitHub. Vào tháng 10/2014, Mark Otto công bố phiên bản Bootstrap 4 đang được phát triển. Phiên bản alpha đầu tiên của Bootstrap 4 đã được triển khai vào tháng 8/2015. Phiên bản mới nhất của Bootstrap được giới thiệu đến người dùng là Bootstrap 4.3.1. Cho đến nay, Bootstrap vẫn là một trong những framework thiết kế website có lượng người dùng “khủng” nhất.

1.5.3. Tại sao nên sử dụng Bootstrap?

Giữa muôn vàn ứng dụng thiết kế website hiện nay, Bootstrap vẫn có khả năng cạnh tranh cao là nhờ những đặc điểm nổi bật sau:

- Dễ dàng thao tác
- Tùy chỉnh dễ dàng
- Chất lượng sản phẩm đầu ra hoàn hảo
- Độ tương thích cao

- Dễ dàng thao tác:

Cơ chế hoạt động của Bootstrap là dựa trên xu hướng mã nguồn mở HTML, CSS và Javascript. Người dùng cần trang bị kiến thức cơ bản 3 mã này mới có thể sử dụng Bootstrap hiệu quả. Bên cạnh đó, các mã nguồn này cũng có thể dễ dàng thay đổi và chỉnh sửa tùy ý.

- Tùy chỉnh dễ dàng:

Bootstrap được tạo ra từ các mã nguồn mở cho phép designer linh hoạt hơn. Giờ đây có thể lựa chọn những thuộc tính, phần tử phù hợp với dự án họ đang theo

đuôi. CDN Bootstrap còn giúp bạn tiết kiệm dung lượng vì không cần tải mã nguồn về máy.

- Chất lượng sản phẩm đầu ra hoàn hảo:

Bootstrap là sáng tạo của các lập trình viên giỏi trên khắp thế giới. Bootstrap đã được nghiên cứu và thử nghiệm trên các thiết bị. Được kiểm tra nhiều lần trước khi đưa vào sử dụng. Do đó, khi chọn Bootstrap, bạn có thể tin rằng mình sẽ tạo nên những sản phẩm với chất lượng tốt nhất.

- Độ tương thích cao:

Điểm cộng lớn nhất của Bootstrap là khả năng tương thích với mọi trình duyệt và nền tảng. Đây là một điều cực kì quan trọng và cần thiết trong trải nghiệm người dùng. Sử dụng Grid System cùng với hai bộ tiền xử lý Less và Sass, Bootstrap mặc định hỗ trợ Responsive và ưu tiên cho các giao diện trên thiết bị di động hơn. Bootstrap có khả năng tự động điều chỉnh kích thước website theo khung browser. Mục đích để phù hợp với màn hình của máy tính để bàn, tablet hay laptop.

1.5.4. Cấu trúc và tính năng của Bootstrap là gì?

- Bootstrap chứa các tập tin JavaScript, CSS và fonts đã được biên dịch và nén lại. Ngoài ra, Bootstrap được thiết kế dưới dạng các mô-đun. Do đó, dễ dàng tích hợp với hầu hết các mã nguồn mở như WordPress, Joomla, Magento, ... Trong đó, Bootstrap mang đến nhiều chức năng nổi bật.

- Bootstrap cho phép người dùng truy cập vào thư viện “khổng lồ” các thành tố dùng để tạo nên giao diện của một website hoàn chỉnh như font, typography, form, table, grid...
- Bootstrap cho phép bạn tùy chỉnh framework của website trước khi tải xuống và sử dụng nó tại website của bạn.
- Tái sử dụng các thành phần lặp đi lặp lại trên website.
- Bootstrap được tích hợp jQuery. Bạn chỉ cần khai báo chính xác các tính năng trong quá trình lập trình web của bạn.
- Định nghĩa glyphicons nhằm giảm thiểu việc sử dụng hình ảnh làm biểu tượng và tăng tốc độ tải trang.

1.6. Giới thiệu về jQuery

1.6.1. Định nghĩa về jQuery

jQuery là một thư viện JavaScript nhỏ gọn, chạy nhanh. jQuery giải quyết các vấn đề tương tác với HTML, bắt sự kiện, hiệu ứng động , v.v. trở nên rất đơn giản.

Mọi tính năng của jQuery thực ra nó bao bọc các lệnh, hàm của JavaScript nên để hiểu jQuery bạn cần hiểu về HTML cơ bản cũng như CSS cơ bản trước.

Các tính năng chính jQuery cung cấp gồm:

- Tương tác với HTML/DOM
- Tương tác với CSS
- Bắt và xử lý sự kiện HTML
- Các hiệu ứng và chuyển động trong HTML
- AJAX (Asynchronous JavaScript and XML)
- JSON parsing
- Các tiện ích xây dựng sẵn

1.6.2. Tại sao nên chọn jQuery

Hiện nay có nhiều JavaScript Framework xuất hiện nhưng jQuery vẫn là Framework phổ biến nhất và nhiều thành phần mở rộng cài thêm (extend). Nhiều công ty công nghệ lớn sử dụng jQuery như Google, Microsoft, IBM, Netflix , v.v.

Ngoài ra jQuery còn tương thích với nhiều trình duyệt khác nhau thậm chí là IE6, bạn thật sự không cần bận tâm nhiều về việc code của mình không chạy được trên các trình duyệt khác nhau.

1.6.3. Cài đặt tích hợp jQuery vào website

Để sử dụng jQuery trong trang HTML, bạn cần đảm bảo trang HTML của mình load chính xác thư viện jQuery bằng cách sử dụng thẻ <script> ở phần <head> của HTML

```
<html>
  <head>
    <script src="file-jquery.js"></script>
  </head>
  <body>
    ....
  </body>
</html>
```

Cách 1 : tải về file js thư viện tại jQuery download (nên chọn bản đã nén có chữ .min.js), sau đó tích hợp vào trang bằng thẻ <script>. Ví dụ: <script src="/public/jquery-3.1.1.min.js"></script>

Cách 2 : thay vì phải tải về file thư viện, ta dùng luôn từ các CDN (Content Delivery Network) giúp cho trang tải nhanh hơn.

Dùng CDN Google : Lấy link phiên bản jQuery tại jQuery Google, sau đó tích hợp vào Website, ví dụ:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">  
</script>
```

1.6.4. Cú pháp jQuery

Khi sử dụng jQuery, bạn cần chọn các phần tử(query) sau đó thực hiện các hành động trên chúng, điều này được thể hiện bằng cú pháp sử dụng jQuery:

\$("selector").action()

- \$ là ký hiệu cho biết bạn truy cập jQuery
- (selector) phần tìm phần tử HTML, theo cách chọn phần tử như đã biết trong CSS (xem thêm chọn phần tử HTML). Ví dụ chọn tất cả các phần tử p là \$('p'), ví dụ chọn phần tử có id là examp là \$('#examp'), ví dụ chọn các phần tử có class là examclass là \$('.examclass')
- action() là các hành động trên phần tử đã chọn, là các hàm mà jQuery cung cấp, bạn sẽ học dần các hàm này

Xem lại ví dụ trên, câu lệnh theo cú pháp đó là

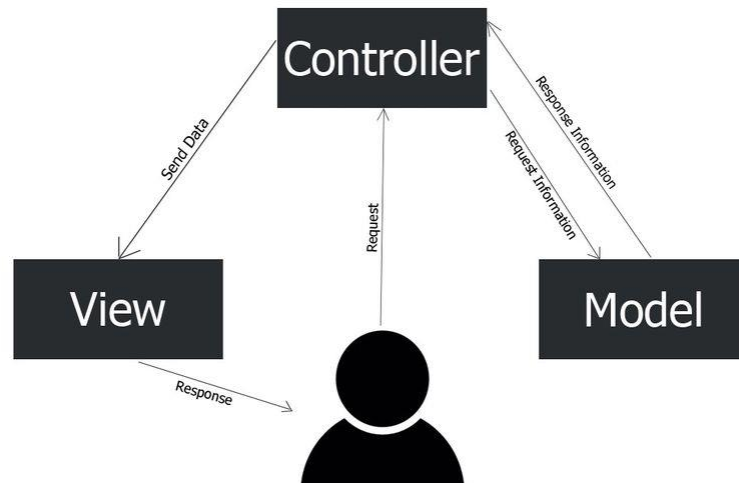
```
$("#start").html("Go!");
```

1.7 Tìm hiểu về mô hình MVC

1.7.1. MVC là gì?

Mô hình MVC (Model – View – Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp cho các developer tách ứng dụng của họ ra 3 thành phần khác nhau Model, View và Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

Model-View-Controller



Hình 1.3: Cấu trúc mô hình MVC

1.7.2. Các thành phần trong MVC

- Model:

Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý...

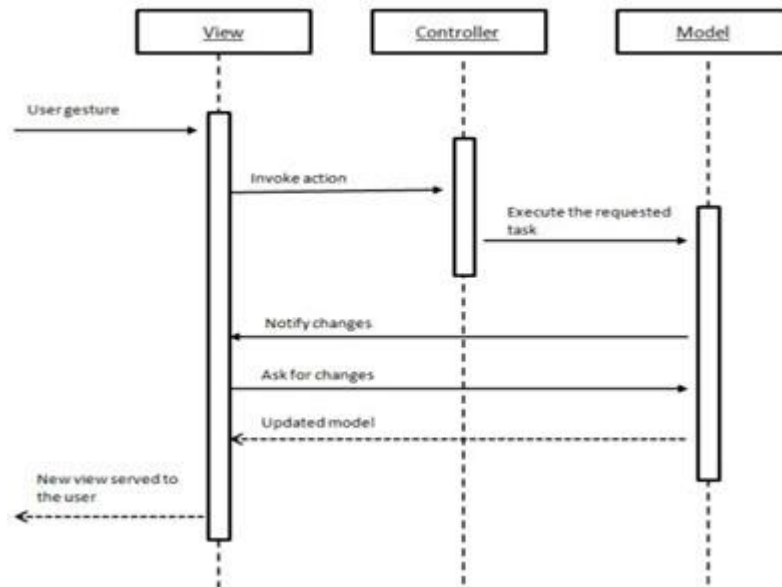
- View:

Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images... Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

- Controller:

Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model.

1.7.3. MVC làm việc như thế nào?



Hình 1.4: Sơ đồ hoạt động của MVC

Nhìn lại sơ đồ phía trên, ta thấy có mũi tên nét liền và những mũi tên nét đứt. Những mũi tên nét đứt được hình thành trên quan điểm của người dùng mà không phải là của những nhà thiết kế phần mềm thực sự. Do đó chúng ta chỉ quan tâm đến những mũi tên còn lại.

Đây là một cách đơn giản để mô tả lại luồng sự kiện được xử lý trong MVC:

- User tương tác với View, bằng cách click vào button, user gửi yêu cầu đi.
- Controller nhận và điều hướng chúng đến đúng phương thức xử lý ở Model.
- Model nhận thông tin và thực thi các yêu cầu.
- Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.

1.7.4. Ưu điểm và nhược điểm của MVC

- *Ưu điểm:*

Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì..

- *Nhược điểm:*

Đối với dự án nhỏ việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

CHƯƠNG II: THIẾT KẾT BACK-END, FRONT-END

1. Phân tích thiết kế hệ thống

Phát biểu bài toán:

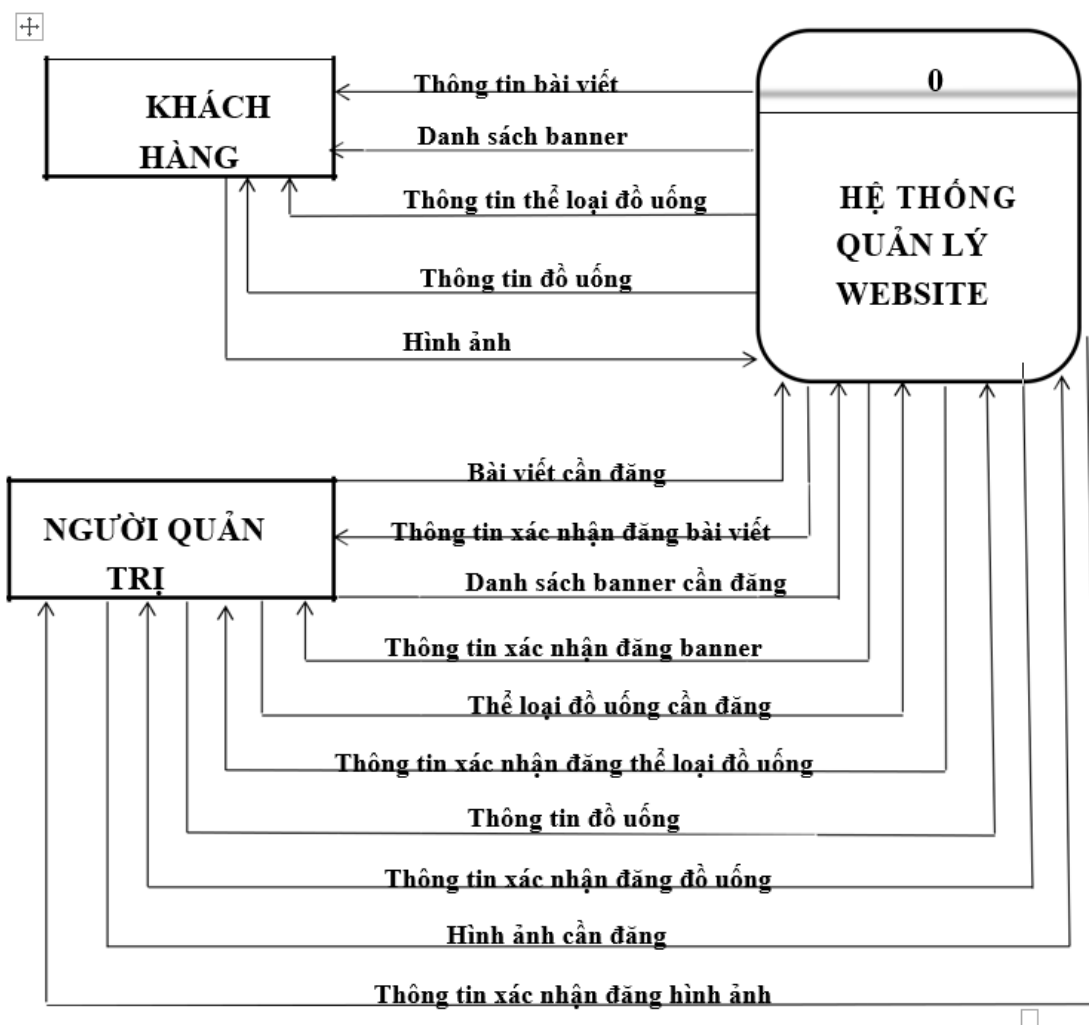
Quán Hellen Tea & Cofffee là nơi kinh doanh đồ uống, được khai trương vào tháng 8 năm 2020. Quán thuộc quyền sở hữu của gia đình, có quy mô khá nhỏ (sức chứa khoảng 30 người). Tại đó bán đa dạng các loại đồ uống như: trà sữa, coffee, sinh tố, v.v. Hellen Tea & Coffee còn nhận tổ chức các sự kiện có quy mô vừa và nhỏ ví dụ như: tiệc, sinh nhật, v.v. Ngoài việc bán hàng trực tiếp tại thì quán còn bán online và giao hàng tận nơi sử dụng dịch vụ trên Foody.vn.

Mục tiêu xây dựng Website cho quán Hellen Tea & Cofffee để quảng bá rộng rãi thông tin của quán, giúp người dùng dễ dàng truy cập để tìm kiếm thông tin và dịch vụ hiện có. Trong đó gồm:

- Trang người dùng giới thiệu được:
 - Thông tin của quán
 - Các sản phẩm, dịch vụ của quán như danh mục đồ uống, nhóm đồ uống, hình ảnh và liên kết để có thể mua sản phẩm trực tuyến sử dụng dịch vụ trên Foody.vn.
 - Các bài viết về quán cũng như các thông tin về sản phẩm dịch vụ.
 - Thông tin liên hệ với quán cho việc đặt bàn, đặt tổ chức các sự kiện, v.v.
 - Các liên kết với các trang thông tin khác của quán để người dùng dễ dàng kết nối như: Fanpage Facebook, Foody, Instagram, v.v.
- Trang quản trị bao gồm các chức năng thêm, sửa, xóa, hiển thị:
 - Đồ uống
 - Thẻ loại đồ uống
 - Hình ảnh
 - Bài viết

Thông qua website này, mọi thông tin, chương trình hoạt động và điểm đặc sắc của sản phẩm được thể hiện một cách chuyên nghiệp, bắt kịp các xu hướng web thịnh hành làm nổi bật những nét riêng của quán. Từ đó thu hút sự quan tâm của khách hàng, người dùng và cả giới truyền thông. Website còn là chất liệu quảng cáo cho nhiều phương thức tiếp thị số khác như quảng cáo Facebook, quảng cáo Google Tìm kiếm, Youtube,... qua đó có thể phần nào giúp nâng cao doanh thu bán hàng của quán trong thời đại 4.0 hiện nay.

Xây dựng biểu đồ ngữ cảnh



Hình 2.1: Biểu đồ ngữ cảnh

Thiết kế hệ thống: Website được chia làm hai thành phần:

- **Front-end** là một phần của website, tương tác trực tiếp với người sử dụng. Cụ thể đó là hệ thống các giao diện người dùng(GUI) và lập trình phía người dùng
- **Back-end** (Lập trình trên server) gồm có các thành phần để xử lý thông tin từ Front-end. Là việc tương tác với hệ quản trị cơ sở dữ liệu.

Quản trị viên của website đều có tài khoản và mật khẩu để đăng nhập vào phần quản trị website. Mỗi tài khoản đều có chức năng phân quyền người dùng được chia làm hai vai trò, vai trò người quản trị sẽ có tất cả các quyền sử dụng các chức năng trong website, vai trò người quản lý sẽ bị hạn chế chức năng hơn.

Quản trị viên tạo các danh mục, các danh mục được sắp xếp trên các thanh menu của website, và được tích hợp chức năng thêm, sửa, xóa, hiển thị.

Quản trị viên truy cập vào chức năng **Danh sách đồ uống** dùng để quản lý thông tin các loại đồ uống, hình ảnh của đồ uống và có các chức năng thêm, sửa, xóa, hiển thị. Các đồ uống được lưu trực tiếp vào trong cơ sở dữ liệu và trong dữ liệu vật lý trên server.

Quản trị viên truy cập vào chức năng **Danh sách Thể loại đồ uống** dùng để quản lý các thể loại đồ uống trên web, có thể thêm, sửa, xóa, hiển thị. Đồ uống là tập con của thể loại đồ uống.

Quản trị viên truy cập vào chức năng **Danh sách Banner** dùng để quản lý những hình ảnh, banner được upload lên làm banner của website hiển thị ở trang chủ và có các chức năng thêm, sửa, xóa, hiển thị. Các ảnh được lưu trực tiếp vào trong cơ sở dữ liệu và trong dữ liệu vật lý trên server.

Quản trị viên truy cập vào chức năng **Danh sách Ảnh** dùng để quản lý tất cả những hình ảnh, banner được upload lên website và có các chức năng thêm, sửa, xóa, hiển thị. Các ảnh được lưu trực tiếp vào trong cơ sở dữ liệu và trong dữ liệu vật lý trên server.

Quản trị viên truy cập vào chức năng **Danh sách Bài viết** dùng để quản lý những bài viết được đăng tải lên website và có các chức năng thêm, sửa, xóa, hiển thị. Các bài viết được lưu trực tiếp vào trong cơ sở dữ liệu và trong dữ liệu vật lý trên server. Bài viết có thể là bài viết nổi bật hoặc không nổi bật. Mỗi bài viết sẽ có ảnh đại diện của bài viết, khi bài viết không có ảnh đại diện sẽ sử dụng ảnh mặc định. Mỗi bài viết có tiêu đề bài viết, nội dung của bài viết, ngày đăng bài, bài viết thuộc nhóm các danh mục nào, và mỗi bài viết có thể có nhiều ảnh bài viết.

Hàng ngày, khách sẽ truy cập vào website của quán để xem thông tin giới thiệu về quán, thông tin các bài viết nổi bật, bài viết mới nhất, thông tin đồ uống, thông tin khuyến mại và thông tin liên hệ để đặt bàn, đặt tổ chức sự kiện và order đồ uống.

1.1. Thiết kế cơ sở dữ liệu

a. Bảng User:

User		
PK	ID	int(11)
	userName	varchar(50)
	password	varchar(50)
	phoneNumber	varchar(50)
	address	varchar(50)
	roleID	int(11)
	dateCreate	timestamp
	status	boolean

Hình 2.2: Bảng User

Mô tả bảng tài khoản (User):

- + Khóa chính là ID, kiểu dữ liệu là integer và sẽ tự tăng khi có tài khoản mới được thêm vào database
- + **userName**, **password**, **phoneNumber** phục vụ cho việc xác thực (Authentication)
- + **roleID** là foreign key và là khóa chính của bảng **Role**, có kiểu dữ liệu là integer phục vụ cho việc phân quyền (Authorization)
- + **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa tài khoản
- + **dateCreate** có kiểu dữ liệu là time chính là thời điểm tài khoản được tạo

b. Bảng *BannerSlider*:

BannerSlider		
PK	ID	Int(11)
	name	varchar(50)
	image1	varchar(255)
	image2	varchar(255)
	image...	varchar(255)
	userID	int(11)
	dateCreate	timestamp
	status	boolean

Hình 2.3: Bảng *BannerSlider*

Mô tả bảng thanh trượt biểu ngữ (*BannerSlider*):

+ Khóa chính là ID, kiểu dữ liệu là integer và sẽ tự tăng khi có Banner Slider mới được thêm vào database

+ **name** là tên của Banner Slider

+ **image** là đường link đến với từng ảnh

+ **userID** là foreign key và là khóa chính của bảng **user**, có kiểu dữ liệu là integer, là id của tài khoản tạo Banner Slider

+ **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa Banner Slider

+ **dateCreate** có kiểu dữ liệu là time chính là thời điểm Banner Slider được tạo

c. Bảng Post:

Post		
PK	ID	int(11)
	name	varchar(50)
	shortDescription	text
	content	text
	bannerID	int(11)
	userID	int(11)
	categoryID	int(11)
	dateCreate	timestamp
	status	boolean

Hình 2.4: Bảng Post

Mô tả bảng bài viết (Post):

+Khóa chính là ID, kiểu dữ liệu là interger và sẽ tự tăng khi có bài viết mới được thêm vào database

+ **name** là tên của bài viết

+ **shortDescription** là mô tả ngắn của bài viết

+ **content** là nội dung của bài viết

+ **bannerId** là id của banner được đặt trên đầu trang và là khóa chính của bảng **BannerSlider**

+ **userID** là foreign key và là khóa chính của bảng **user**, có kiểu dữ liệu là interger, là id của tài khoản tạo bài viết

+ **categoryID** là foreign key và là khóa chính của bảng **Category**, có kiểu dữ liệu là interger, là id của thể loại của bài viết

+ **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa bài viết

+ **dateCreate** có kiểu dữ liệu là time chính là thời điểm bài viết được tạo

d. Bảng Drink:

Drink		
PK	ID	int(11)
	name	varchar(50)
	description	text
	price	float
	discountPrice	float
	userID	int(11)
	categoryID	int(11)
	dateCreate	timestamp
	status	boolean

Hình 2.5: Bảng Drink

Mô tả bảng đồ uống (Drink):

+ Khóa chính là ID, kiểu dữ liệu là interger và sẽ tự tăng khi có đồ uống mới được thêm vào database

+ **name** là tên của đồ uống

+ **description** là mô tả của đồ uống

+ **price** là giá của đồ uống

+ **discountPrice** là giá khuyến mại của đồ uống

+ **userID** là foreign key và là khóa chính của bảng **user**, có kiểu dữ liệu là interger, là id của tài khoản tạo đồ uống

+ **categoryID** là foreign key và là khóa chính của bảng **Category**, có kiểu dữ liệu là interger, là id của thể loại của đồ uống

+ **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa đồ uống

+ **dateCreate** có kiểu dữ liệu là time chính là thời điểm đồ uống được thêm

e. Bảng Image:

Image		
PK	ID	int(11)
	name	varchar(50)
	source	varchar(255)
	imageCategoryID	int(11)
	dateCreate	timestamp
	status	boolean

Hình 2.6: Bảng Image

Mô tả bảng hình ảnh (Image):

+ Khóa chính là ID, kiểu dữ liệu là interger và sẽ tự tăng khi có hình ảnh mới được thêm vào database

+ **name** là tên của ảnh

+ **source** là đường link của ảnh

+ **imageCategoryID** là foreign key và là khóa chính của bảng **Category**, có kiểu dữ liệu là interger, là id của thể loại của ảnh (có thể là ảnh của đồ uống hoặc ảnh bài viết)

+ **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa đồ uống

+ **dateCreate** có kiểu dữ liệu là time chính là thời điểm hình ảnh được thêm

f. Bảng Category:

Category		
PK	ID	int(11)
	name	varchar(50)
	shortDescription	text
	status	boolean

Hình 2.7: Bảng Category

Mô tả bảng thể loại đồ uống (**Category**):

+Khóa chính là ID, kiểu dữ liệu là interger và sẽ tự tăng khi có thể loại mới được thêm vào database.

Ngoài ra còn có các thông tin như:

- + **name** là tên của thể loại đồ uống
- + **shortDescription** là mô tả ngắn về thể loại đồ uống
- + **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa thể loại.

g. Bảng Role:

Role		
PK	ID	int(11)
	roleName	varchar(50)
	description	text
	status	boolean

Hình 2.8: Bảng Role

Mô tả bảng vai trò (**Role**):

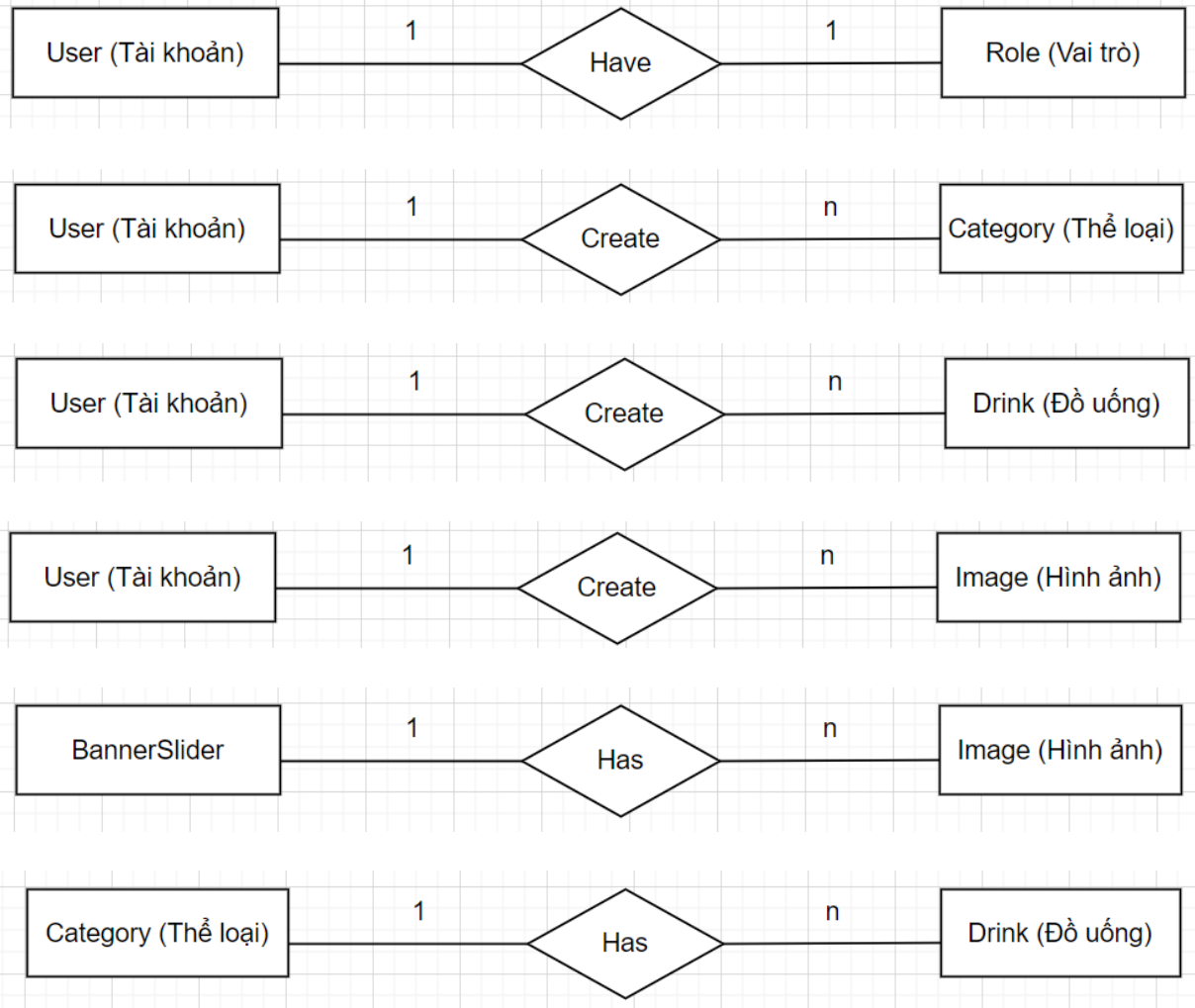
+ Khóa chính là ID, kiểu dữ liệu là interger và sẽ tự tăng khi có vai trò mới được thêm vào database.

Ngoài ra còn có các thông tin như:

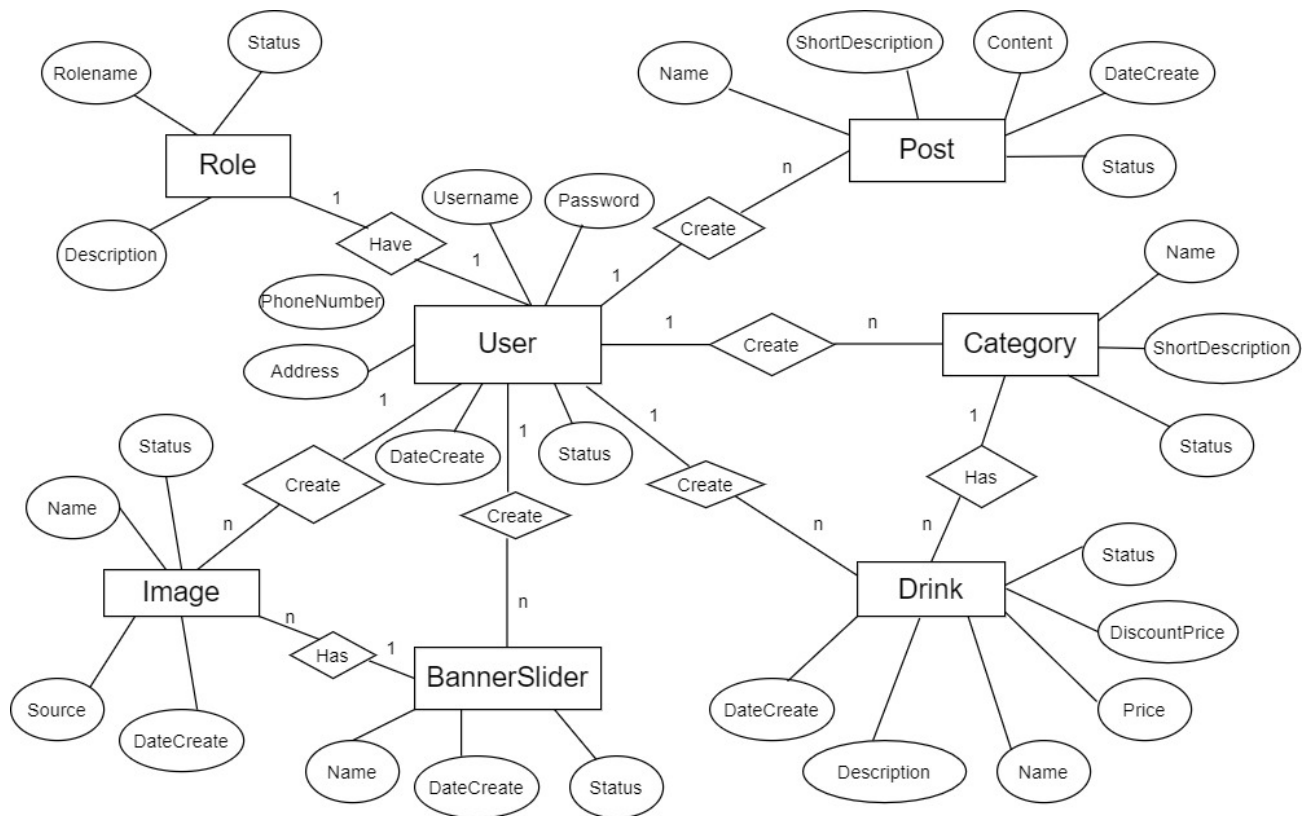
- + **roleName** là tên của vai trò
- + **description** là mô tả về vai trò

+ **status** có kiểu dữ liệu boolean cho phép chỉnh để kích hoạt hoặc vô hiệu hóa vai trò.

1.2. Lược đồ quan hệ

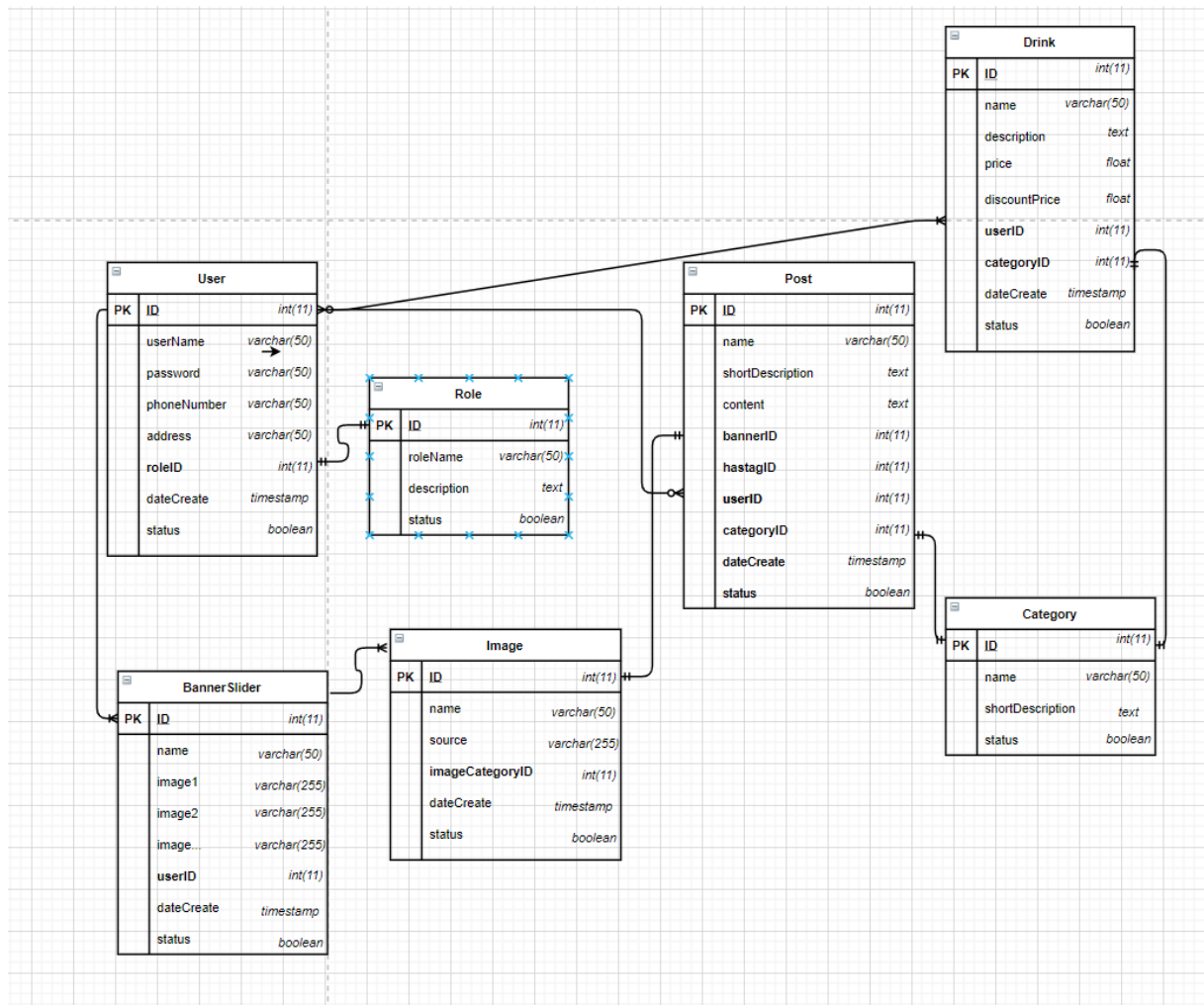


1.3. Mô hình thực thể E-R



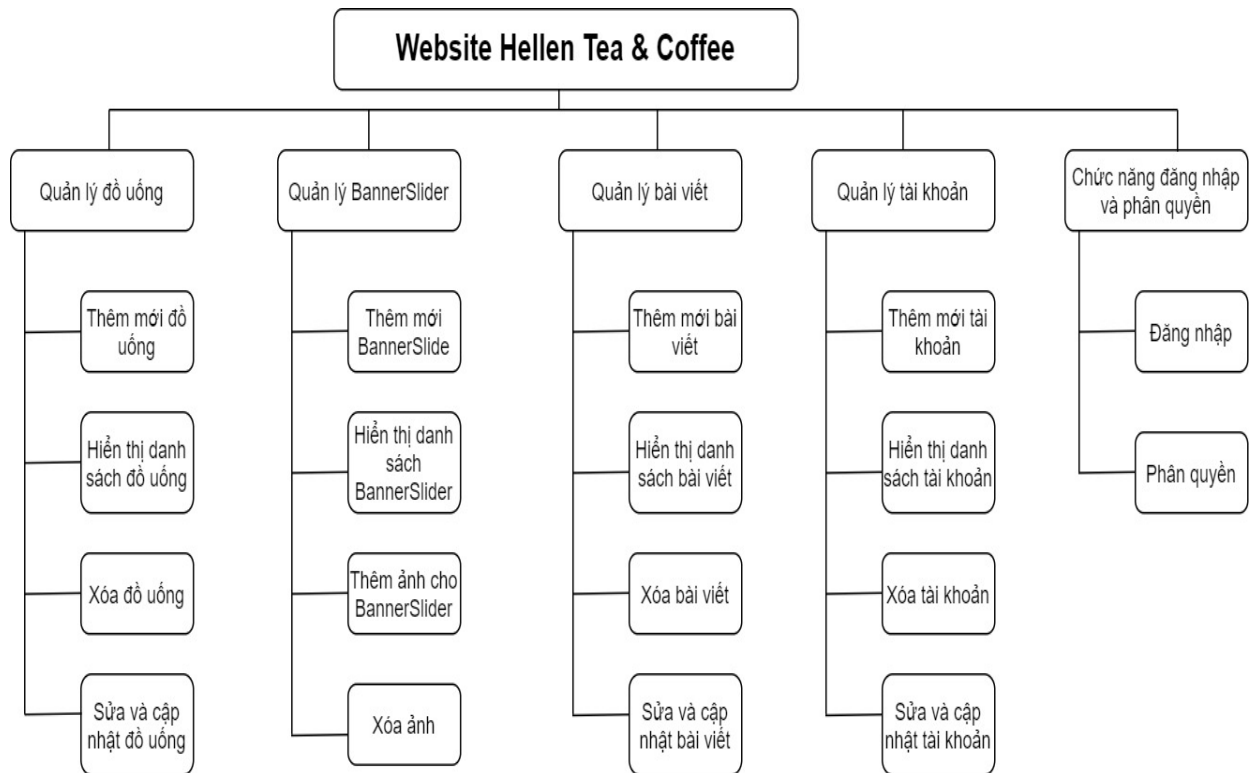
Hình 2.9 : Mô hình thực thể E-R

1.4. Mô hình liên kết dữ liệu



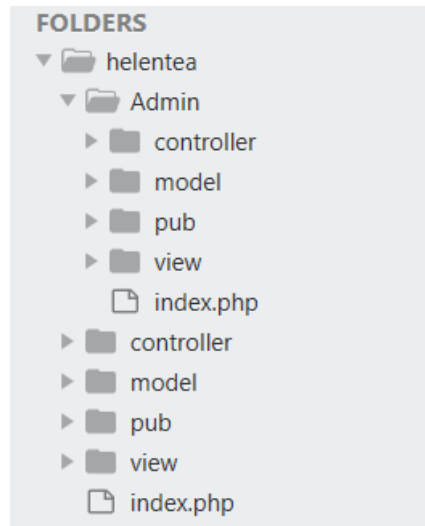
Hình 2.10 : Mô hình liên kết dữ liệu

1.5. Sơ đồ phân rã chức năng



Hình 2.11: Sơ đồ phân rã chức năng

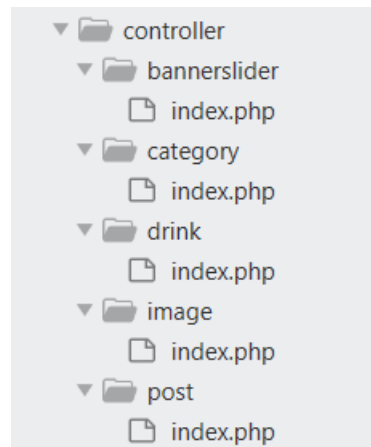
2. Phân tích cấu trúc thư mục



Hình 2.12: Cấu trúc thư mục

- Model, View và Controller của trang quản lý (Admin) sẽ nằm trong thư mục Admin.
- Model, View và Controller nằm ngoài thư mục Admin là của trang người dùng.

2.1. Controller

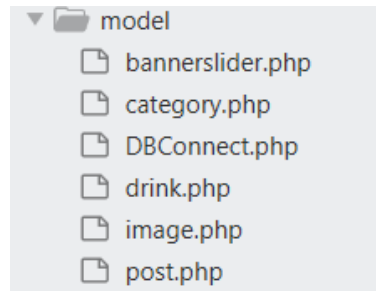


Hình 2.13: Cấu trúc thư mục Controller

Controller có chức năng nhận và xử lý những yêu cầu hoặc dữ liệu được gửi từ người dùng bằng phương thức GET hoặc POST. Yêu cầu sẽ được gửi vào các file ***index.php*** sau đó sẽ được xử lý và thực thi.

Controller sẽ lấy các dữ liệu được gửi tới bằng các phương thức GET hoặc POST bằng cách dùng **`$_GET['key']`** và **`$_POST['key']`**.

2.2. Model



Hình 2.14: Cấu trúc thư mục Model

- Model là những đối tượng như đồ uống, bài viết chứa các hàm (function) để thực thi các câu lệnh sql cho phép thêm, xóa và sửa cơ sở dữ liệu.

- Những câu lệnh để kết nối với database nằm trong file **DBConnect.php** trong thư mục Model. **DBConnect.php** có các hàm như **connect()** để kết nối database hoặc **execute(\$sql)** để thực thi các câu lệnh '\$sql' được truyền vào:

```
DBConnect.php
1 <?php
2 class Database{
3     public $hostName = 'localhost';
4     public $userName = 'root';
5     public $pass = '';
6     public $dbname = 'helentea';
7     public $conn = null;
8     public $result = null;
9
10    public function connect(){
11        $this->conn = new mysqli($this->hostName, $this->userName, $this->pass, $this->dbname);
12        if (!$this->conn) {
13            echo "Ket noi database that bai";
14        }else{
15            mysqli_set_charset($this->conn, 'utf8');
16        }
17        return $this->conn;
18    }
19
20    public function execute($sql){
21        $this->result = $this->conn->query($sql);
22        return $this->result;
23    }
24 }
```

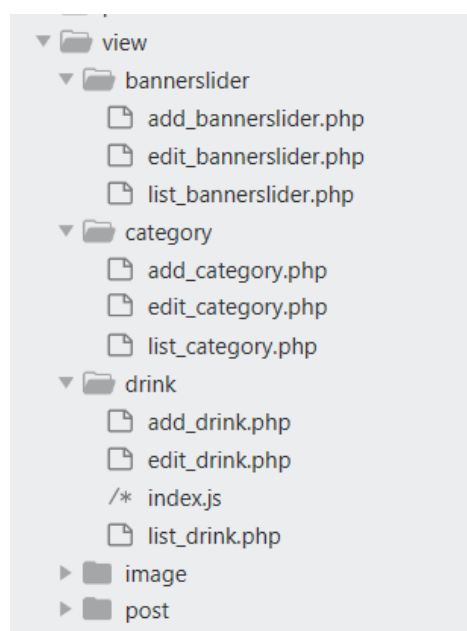
Hình 2.15: Mã nguồn file DBConnect.php

Ngoài ra còn các hàm khác để lấy dữ liệu từ các bảng trong database:

```
DBConnect.php
25 //lay 1 row
26 public function getData()
27 {
28     if ($this->result) {
29         $data = mysqli_fetch_array($this->result);
30     }else{
31         $data = 0;
32     }
33     return $data;
34 }
35
36 //lay 1 row
37 public function getDataByID($table, $id)
38 {
39     $sql = "SELECT * FROM $table where ID = $id";
40     $this->execute($sql);
41     if ($this->countRow() != 0) {
42         $data = mysqli_fetch_array($this->result);
43     }else{
44         $data = 0;
45     }
46     return $data;
47 }
48
49 //lay ca table
50 public function getAllData($table){
51     $sql = "SELECT * FROM $table";
52     $this->execute($sql);
53     if ($this->countRow() == 0) {
54         $data = 0;
55     }else{
56         while ($datas = $this->getData()) {
57             $data[] = $datas;
58         }
59     }
60     return $data;
61 }
```

Hình 2.16: Mã nguồn file DBConnect.php

2.3. View

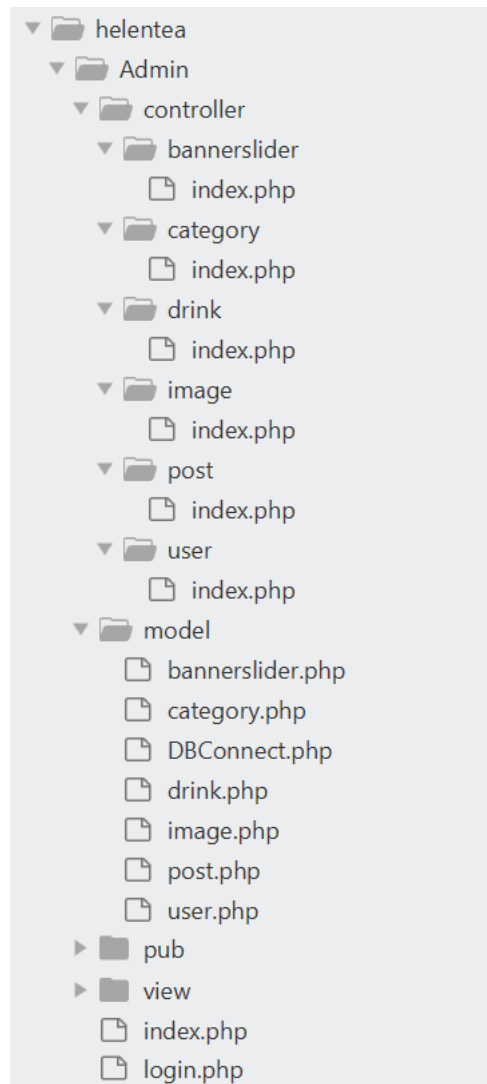


Hình 2.17: Cấu trúc thư mục View

View là các mã HTML, CSS và JS tạo nên giao diện mà người dùng sẽ nhìn thấy (Front-end).

3. Xây dựng Back-end

- Do website được xây dựng theo mô hình MVC nên mỗi chức năng sẽ đều có đủ các nhân tố Model, View và Controller.



Hình 2.18: Cấu trúc chi tiết thư mục Back-end

3.1. Các chức năng quản lý đồ uống

a. Chức năng thêm mới đồ uống

LOGO	
User picture User Account Status	Thêm mới đồ uống Tên đồ uống Tên đồ uống Ảnh Chọn tệp... Mô tả Mô tả Giá Giá Thể loại Chọn thể loại Thêm mới đồ uống

Hình 2.19: Thiết kế giao diện chức năng thêm mới đồ uống

Biểu mẫu này được tạo ra bằng cách dùng các thẻ **<form>**, **<input>** và **<table>**. Quản lý sẽ dùng form này để điền các thông tin cho sản phẩm mới.

```
add_drink.php
1 <div class="col-md-6 panel panel-default add-drink">
2 <h3>Thêm mới đồ uống</h3>
3 <form id="add_drink" action="index.php?controller=drink&action=add" method="POST" enctype="multipart/form-data" style="padding: 4%">
4 <table>
5 <tr>
6 <td>Tên đồ uống</td>
7 <td><input class="form-control input-md" type="text" name="name" placeholder="Tên đồ uống..."></td>
8 </tr>
9 <tr>
10 <td>Ảnh</td>
11 <td><input class="form-control input-md" type="file" name="image" placeholder="Chọn ảnh đi cha nội..."></td>
12 </tr>
13 <tr>
14 <td>Mô tả</td>
15 <td><textarea class="form-control input-md" name="description" rows="10" cols="30" placeholder="Mô tả đồ uống..."></td>
16 </tr>
17 <tr>
18 <td>Giá</td>
19 <td><input class="form-control input-md" type="text" name="price" placeholder="Nhập giá..."></td>
20 </tr>
21 <tr>
22 <td>Giá khuyến mãi</td>
23 <td><input class="form-control input-md" type="text" name="discountPrice" placeholder="Nhập giá khuyến mãi..."></td>
24 </tr>
25 <tr>
26 <td>Người nhập</td>
27 <td><input class="form-control input-md" type="text" name="userID"></td>
28 </tr>
29 <tr>
30 <td>Thể loại</td>
31 <td><select class="form-control input-md" name="categoryID">
32 <?php
33 <foreach ($categories as $category) {
34 <?>
35 <option value="<?=$category['ID']"><?=$category['name']"></option>
36 <?php
37 <?>
38 </foreach>
39 </td>
40 </tr>
41 <tr>
42 <td>Trạng thái</td>
43 <td><select class="form-control input-md" name="status">
44 <option value="1" selected="selected">Enable</option>
45 <option value="0">Disable</option>
46 </select>
47 </td>
48 </tr>
49 <tr>
50 <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="add_drink"></td>
51 </tr>
52 </table>
53 </div>
```

Hình 2.20: Mã nguồn giao diện chức năng thêm mới đồ uống

Dưới đây là controller xử lý những yêu cầu về đồ uống, cụ thể là chức năng thêm mới đồ uống. Controller sẽ lấy các dữ liệu về đồ uống mà người dùng gửi lên qua form và chạy hàm **Insert Data** để thêm đồ uống mới vào cơ sở dữ liệu. **Switch ...case** được sử dụng để lọc các yêu cầu từ người dùng như add, list,, v.v.

```
index.php x list_drink.php x
1 <?php
2 include "model/drink.php";
3 $db = new Drink;
4 $db->connect();
5 $tblTable = 'drink';
6 $subTable1 = 'category';
7 $categories = $db->getAllData($subTable1);
8 if (isset($_GET['action'])) {
9     $action = $_GET['action'];
10 }else{
11     $action = '';
12 }
13 switch ($action) {
14     case 'add':{
15         if (isset($_POST['add_drink'])) {
16             $name = $_POST['name'];
17             $description = $_POST['description'];
18             $price = $_POST['price'];
19             $discountPrice = $_POST['discountPrice'];
20             $userID = $_POST['userID'];
21             $categoryID = $_POST['categoryID'];
22             $status = $_POST['status'];
23             $image = $_FILES['image']['name'];
24             if (isset($_FILES['image'])) {
25                 move_uploaded_file($_FILES['image']['tmp_name'], 'pub/image/' . $_FILES['image']['name']);
26             }
27             $db->InsertData($name, $image, $description, $price, $discountPrice, $userID, $categoryID, $status);
28             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=drink&action=list';</script>");
29         }
30         require_once('view/drink/add_drink.php');
31         break;
32     }
33     case 'list':{
34         if ($db->getAllData($tblTable)) {
35             $data = $db->getAllData($tblTable);
36         } else $data = null;
37         require_once('view/drink/list_drink.php');
38         break;
39     }
40 }
41
42
43
44
```

Hình 2.21: Mã nguồn chức năng thêm mới đồ uống

Và đây là hàm **InsertData()** trong model nhận các dữ liệu được gửi từ controller và thực thi câu lệnh **sql (execute())** để thêm mới đồ uống vào cơ sở dữ liệu. Ngoài ra, câu lệnh **move_uploaded_file()** được sử dụng để lưu ảnh đã tải lên vào thư mục, cụ thể ở đây là thư mục pub/image.

```
<?php
include "DBConnect.php";
/**
 *
 */
class Drink extends Database
{
    //them du lieu
    public function InsertData($name, $image, $description, $price, $discountPrice, $userID, $categoryID, $status){
        $sql = "INSERT INTO drink(name, image, description, price, discountPrice, userID, categoryID, status)
            VALUES ('$name', '$image', '$description', '$price', '$discountPrice', '$userID', '$categoryID', '$status')";
        var_dump($sql);
        return $this->execute($sql);
    }

    //sua du lieu
    public function UpdatetData($id, $name, $image, $description, $price, $discountPrice, $userID, $categoryID, $status){
        $sql = "UPDATE drink SET name='$name', image='$image', description='$description', price='$price', discountPrice='$discountPrice', userID='$userID', categoryID='$categoryID', status='$status' WHERE ID = '$id'";
        return $this->execute($sql);
    }
}
?>
```

Hình 2.22: Mã nguồn chức năng thêm mới đồ uống

b. Chức năng hiển thị danh sách đồ uống

LOGO																																										
User picture User Account Status Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out	Danh sách đồ uống <table border="1"> <thead> <tr> <th>ID</th> <th>Tên đồ uống</th> <th>Loại</th> <th>Mô tả</th> <th>Ảnh</th> <th>Giá</th> <th>Hoạt động</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td> </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td> </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td> </td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td> </td> </tr> </tbody> </table> Thêm mới đồ uống							ID	Tên đồ uống	Loại	Mô tả	Ảnh	Giá	Hoạt động																												
	ID	Tên đồ uống	Loại	Mô tả	Ảnh	Giá	Hoạt động																																			

Hình 2.23: Thiết kế giao diện chức năng hiển thị danh sách đồ uống

Hàm **getAllData('tableName')** được dùng để thực thi câu lệnh sql: **SELECT * FROM tableName** để lấy tất cả các dữ liệu trong bảng **'tableName'**.

```

case 'list':{
    if ($db->getAllData($tblTable)) {
        $data = $db->getAllData($tblTable);

    } else $data = null;

    require_once('view/drink/list_drink.php');
    break;
}

```

Hình 2.24: Mã nguồn chức năng lấy dữ liệu trong bảng

Vòng lặp foreach sẽ được dùng để in tất cả các sản phẩm có trong database ra màn hình:

```
if (isset($data)) {
    foreach ($data as $value) {
        ?>

        <tr style="border-bottom: 1px solid black;">
            <td><?= $value['ID'] ?></td>
            <td><?= $value['name'] ?></td>
            <td>
                <?php
                foreach ($categories as $category) {
                    if ($category['ID'] == $value['categoryID']) {
                        echo $category['name'] ;
                    }
                }
                ?>
            </td>
            <td><?= $value['description'] ?></td>
            <td></td>
            <td><?= $value['price'] ?></td>
            <td><?= $value['discountPrice'] ?></td>
            <td><?= $value['userID'] ?></td>
            <td><?= $value['dateCreate'] ?></td>
            <td><?= $value['status'] ?></td>
            <td>

```

Hình 2.25: Mã nguồn chức năng hiển thị danh sách đồ uống

c. Chức năng xóa đồ uống

ID	Tên đồ uống	Loại	Mô tả	Ảnh	Giá	Hoạt động
						 
						 

Hình 2.26: Thiết kế giao diện chức năng xóa đồ uống

Khi người dùng bấm vào biểu tượng thùng rác, yêu cầu xóa và ID của sản phẩm tương ứng sẽ được gửi đi. Khi controller nhận được yêu cầu xóa từ người dùng và ID được lấy bằng phương thức GET, hàm **DeleteDatabyID()** sẽ được gọi ra để xóa hàng trong bảng với ID tương ứng.

Ngoài ra, câu lệnh **unlink()** còn được dùng để xóa luôn ảnh trong thư mục.

Ảnh dưới đây là đoạn code trong controller phục vụ các tác vụ trên:

```
case 'delete':{
    $id = $_GET['ID'];
    chmod('pub/image/', 0777);
    if ($db->getDataByID($tblTable, $id)['image']) {
        unlink('pub/image/' . $db->getDataByID($tblTable, $id)['image']);
    }
    $db->DeleteDataByID($tblTable, $id);
    $data = $db->getAllData($tblTable);
    require_once('view/drink/list_drink.php');
    break;
}
```

Hình 2.27: Mã nguồn chức năng xóa đồ uống

Đây là hàm **DeleteDataByID()** có chức năng gọi và thực thi câu lệnh sql nằm trong phần model:

```
//lay 1 row
public function DeleteDataByID($table, $id)
{
    $sql = "DELETE FROM $table WHERE ID = $id";
    return $this->execute($sql);
}
```

Hình 2.28: Mã nguồn chức năng xóa đồ uống trong cơ sở dữ liệu

d. Chức năng sửa và cập nhật đồ uống

Danh sách đồ uống						
ID	Tên đồ uống	Loại	Mô tả	Ảnh	Giá	Hoạt động
						 
						 

Hình 2.29: Thiết kế giao diện chức năng sửa và cập nhật đồ uống

Khi người dùng bấm vào biểu tượng bút chì, trang cập nhật thông tin đồ uống sẽ được chuyển tới với các thông tin và ID của hàng tương ứng.

The screenshot shows a web interface for editing a drink. At the top left is a 'LOGO' placeholder. Below it is a sidebar with a 'User Account Status' section containing a 'User picture' icon and a list of links: 'Danh sách đồ uống', 'Danh sách thể loại đồ uống', 'Danh sách Banner', 'Danh sách ảnh', 'Danh sách bài viết', 'Danh sách tài khoản', and 'Log out'. The main content area is titled 'Sửa đồ uống' (Edit Drink). It contains several input fields: 'Tên đồ uống' (Drink Name) with the value 'Hồng trà bạc hà', 'Ảnh' (Image) with a 'Chọn tệp...' (Choose file...) button, 'Mô tả' (Description) with a large text area containing 'Mô tả', 'Giá' (Price) with the value '29000', 'Giá khuyến mại' (Promotional Price) with the value 'Giá khuyến mại', 'Thể loại' (Category) with the value 'Trà hoa quả', and 'Trạng thái' (Status) with the value 'Enable'. At the bottom right of the form is a 'Xác nhận' (Confirm) button.

Hình 2.30: Thiết kế giao diện sửa và cập nhật đồ uống

Để thực hiện được các hành động đã nêu bên trên, đây là đoạn code trong controller sẽ được chạy:

```
case 'edit':{  
    if (isset($_GET['ID'])) {  
        $id = $_GET['ID'];  
        $drinkByID = $db->getDataByID($tblTable,$id);  
    }  
  
    require_once('view/drink/edit_drink.php');  
    break;  
}
```

Hình 2.31: Mã nguồn chức năng sửa và cập nhật đồ uống

Khi yêu cầu ‘edit’ được gửi đi và có bao gồm ID, các thông tin của đồ uống tương ứng sẽ được gọi ra bằng hàm **getDataByID()**.

Trong phần view, một form tương tự với phần thêm mới sản phẩm được tải nhưng những giá trị của các trường đã được gắn sẵn tương ứng với dữ liệu của sản phẩm được yêu cầu sửa.

```
<form id="submit" action="index.php?controller=drink&action=update_drink&ID=<?=$drinkByID['ID']?>" method="POST" enctype="multipart/form-data" style="padding: 4%">
  <table>
    <tr>
      <td>Tên đồ uống</td>
      <td><input class="form-control input-md" type="text" name="name" placeholder="Tên đồ uống..." value="<?=$drinkByID['name']?>"></td>
    </tr>
    <tr>
      <td>Ảnh</td>
      <td>
        <input class="form-control input-md" type="file" name="updateImage" placeholder="Chọn ảnh di cha nội..." >
        <input class="form-control input-md" style="display: none" type="text" name="image" value="<?=$drinkByID['image']?>">
        
      </td>
    </tr>
    <tr>
      <td>Mô tả</td>
      <td><textarea class="form-control input-md" name="description" rows="10" cols="30" placeholder="Mô tả đồ uống...">
        <?=$drinkByID['description']?>
      </textarea></td>
    </tr>
    <tr>
      <td>Giá</td>
      <td><input class="form-control input-md" type="text" name="price" placeholder="Nhập giá..." value="<?=$drinkByID['price']?>"></td>
    </tr>
    <tr>
      <td>Giá khuyến mãi</td>
      <td><input class="form-control input-md" type="text" name="discountPrice" placeholder="Nhập giá khuyến mãi..." value="<?=$drinkByID['discountPrice']?>"></td>
    </tr>
    <tr>
      <td>Người nhập</td>
      <td><input class="form-control input-md" type="text" name="userID" placeholder="Nhập giá khuyến mãi..." value="<?=$drinkByID['userID']?>"></td>
    </tr>
    <tr>
      <td>Thế loại</td>
      <td><select class="form-control input-md" name="categoryID">
        <?php
          foreach ($categories as $category) {
            <?>
            <option value="<?=$category['ID']?>" <?=$category['name']?></option>
          }
        <?php
      </td>
    </tr>
  </table>
</form>
```

Hình 2.32: Mã nguồn giao diện sửa và cập nhật đồ uống

Sau khi chỉnh sửa xong, nếu người dùng bấm vào nút gửi trong form edit, yêu cầu **'update_drink'** sẽ được gửi đi. Nếu người dùng có thay đổi ảnh, ảnh cũ của sản phẩm trong thư mục `pub/image` sẽ bị xóa và thay vào đó là ảnh mới.

```
case 'update_drink':{
    if (isset($_POST['update_drink'])) {
        $id = $_GET['ID'];
        $name = $_POST['name'];
        $description = $_POST['description'];
        $price = $_POST['price'];
        $discountPrice = $_POST['discountPrice'];
        $userID = $_POST['userID'];
        $image = $_POST['image'];
        $categoryID = $_POST['categoryID'];
        if ($_FILES['updateImage']['name']){
            move_uploaded_file($_FILES['updateImage']['tmp_name'], 'pub/image/'.$_FILES['updateImage']['name']);
            if (isset($_POST['image'])){
                unlink('pub/image/'.$_POST['image']);
            }
            $image = $_FILES['updateImage']['name'];
        }

        $status = $_POST['status'];
        $db->UpdateData($id, $name, $image, $description, $price, $discountPrice, $userID, $categoryID, $status);
        echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=drink&action=list';</script>");
    }
    break;
}
```

Hình 2.33: Mã nguồn chức năng sửa và cập nhật đồ uống

Sau khi nhận được dữ liệu được gửi lên từ form **update_drink** bằng phương thức POST (`if(isset($_POST['update_drink']))`), hàm **UpdateData()** sẽ được gọi ra để cập nhật những thông tin người dùng chỉnh sửa vào trong database.

Ngoài ra câu lệnh **move_upload_file()** được dùng để xóa ảnh của sản phẩm đó trong thư mục **pub/image**.

Đây là hàm **UpdateData()** nằm trong model có chức năng thực thi câu lệnh sql Update để cập nhật những thông tin người dùng chỉnh sửa vào trong database.

```
//sua du lieu
public function UpdateData($id, $name, $image, $description, $price, $discountPrice, $userID, $categoryID, $status){
    $sql = "UPDATE drink SET name='$name', image='$image', description='$description', price='$price', discountPrice='$discountPrice', userID='$userID',
        categoryID='$categoryID', status='$status' WHERE ID = '$id'";
    return $this->execute($sql);
}
```

Hình 2.34: Mã nguồn chức năng sửa và cập nhật đồ uống vào cơ sở dữ liệu

3.2 Các chức năng quản lý Banner Slider

a. Chức năng thêm mới Banner Slider

LOGO	
User picture User Account Status	Thêm mới Banner Slider Tên BannerSlider Nhập tên Trạng thái Enable Xác nhận
Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out	

Hình 2.35: Thiết kế giao diện thêm mới Banner Slider

```
1 <div class="col-md-6 panel panel-default add-bannerslider">
2   <h3>Thêm mới bannerslider</h3>
3   <form id="add_bannerslider" action="index.php?controller=bannerslider&action=add" method="POST" enctype="multipart/form-data" style="padding: 4px">
4     <table>
5       <tr>
6         <td>Tên banner slider</td>
7         <td><input class="form-control input-md" type="text" name="name" placeholder="Tên bannerslider..."></td>
8       </tr>
9
10      <tr>
11        <td>Trạng thái</td>
12        <td>
13          <select class="form-control input-md" name="status">
14            <option value="1" selected="selected">Enable</option>
15            <option value="0">Disable</option>
16          </select>
17        </td>
18      </tr>
19      <tr>
20        <td></td>
21        <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="add_bannerslider"></td>
22      </tr>
23    </table>
24  </form>
25 </div>
26
27 <script type="text/javascript">
28 $(function() {
29   $("#submit").click(function(){
30     if (confirm("Xác nhận")){
31       $('#form#add_bannerslider').submit();
32     }
33   });
34 });
```

Hình 2.36: Mã nguồn giao diện thêm mới Banner Slider

Sau khi dữ liệu được gửi lên từ form **add_bannerslider**, hàm **InsertData()** sẽ được gọi để nhập những sản phẩm mới vào database.

```
13 switch ($action) {
14     case 'add':{
15         if (isset($_POST['add_bannerslider'])) {
16             $name = $_POST['name'];
17             $status = $_POST['status'];
18             $db->InsertData($name, $status);
19
20             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=bannerslider&action=list';</script>");
21         }
22         require_once('view/bannerslider/add_bannerslider.php');
23         break;
24     }
25
26     case 'list':{
27         if ($db->getAllData($tblTable)) {
28             $data = $db->getAllData($tblTable);
29         } else $data = null;
30
31         require_once('view/bannerslider/list_bannerslider.php');
32         break;
33     }
34
35     case 'edit':{
36         if (isset($_GET['ID'])) {
37             $id = $_GET['ID'];
38             $bannersliderByID = $db->getDataByID($tblTable,$id);
39         }
40
41         require_once('view/bannerslider/edit_bannerslider.php');
42         break;
43     }
44
45     case 'update_bannerslider':{
46         if (isset($_POST['update_bannerslider'])) {
47             $id = $_GET['ID'];
48             $name = $_POST['name'];
49             $description = $_POST['description'];
50             $bannersliderID = $_POST['bannersliderID'];
51             $status = $_POST['status'];
52             $db->updateData($id, $name, $description, $status);
53         }
54         break;
55     }
56
57
58     case 'delete':{
59         $id = $_GET['ID'];
60         $db->deleteDataByID($tblTable, $id);
61         $data = $db->getAllData($tblTable);
62         require_once('view/bannerslider/list_bannerslider.php');
63         break;
64     }
65 }
66 >>
```

Hình 2.37: Mã nguồn chức năng thêm mới Banner Slider

Và đây là hàm **InsertData()** trong model có chức năng thực thi câu lệnh thêm mới bannerslider trong database.

```
//them du lieu
public function InsertData($name, $status){
    $sql = "INSERT INTO bannerslider(name, status)
           VALUES ('$name', '$status')";
    return $this->execute($sql);
}
```

Hình 2.38: Mã nguồn chức năng thêm mới Banner Slider vào cơ sở dữ liệu

b. Chức năng hiển thị danh sách cho Banner Slider

LOGO																										
User picture User Account Status	Danh sách Banner Slider <table border="1"><thead><tr><th>ID</th><th>Tên banner slider</th><th>Danh sách ảnh</th><th>Trạng thái</th><th>Hoạt động</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td><td> </td></tr><tr><td></td><td></td><td></td><td></td><td> </td></tr><tr><td></td><td></td><td></td><td></td><td> </td></tr><tr><td></td><td></td><td></td><td></td><td> </td></tr></tbody></table> Thêm mới Banner Slider	ID	Tên banner slider	Danh sách ảnh	Trạng thái	Hoạt động																				
ID	Tên banner slider	Danh sách ảnh	Trạng thái	Hoạt động																						

Hình 2.39: Thiết kế giao diện danh sách Banner Slider

Danh sách Banner Slider sẽ được hiển thị dưới dạng bảng.

Để có các dữ liệu hiển thị ra bên ngoài front end, hàm **getAllData()** sẽ được gọi để lấy hết các dữ liệu có trong database theo tên bảng được truyền vào:

```
case 'list':{  
    if ($db->getAllData($tblTable)) {  
        $data = $db->getAllData($tblTable);  
    } else $data = null;  
  
    require_once('view/bannerslider/list_bannerslider.php');  
    break;  
}
```

Hình 2.40: Mã nguồn chức năng hiển thị danh sách Banner Slider

```
//lay ca table
public function getAllData($table){
    $sql = "SELECT * FROM $table";
    $this->execute($sql);
    if ($this->countRow() == 0) {
        $data = 0;
    }else{
        while ($datas = $this->getData()) {
            $data[] = $datas;
        }
    }
    return $data;
}
```

Hình 2.41: Mã nguồn chức năng hiển thị danh sách Banner Slider

Với phần hiển thị danh sách Banner Slider, vòng lặp foreach được dùng để có thể hiển thị tất cả các dòng có trong bảng:

```
<?php
if (isset($data)) {
    foreach ($data as $value) {
        ?>
        <tr style="border-bottom: 1px solid black;">
            <td><?= $value['ID'] ?></td>
            <td><?= $value['name'] ?></td>
            <td>
                <?php
                foreach ($images as $image) {
                    if ($image['bannerID'] == $value['ID'] ){
                        ?>
                        
                    <?php
                }
            }
        }
    }
}
```

Hình 2.42: Mã nguồn chức năng hiển thị danh sách Banner Slider

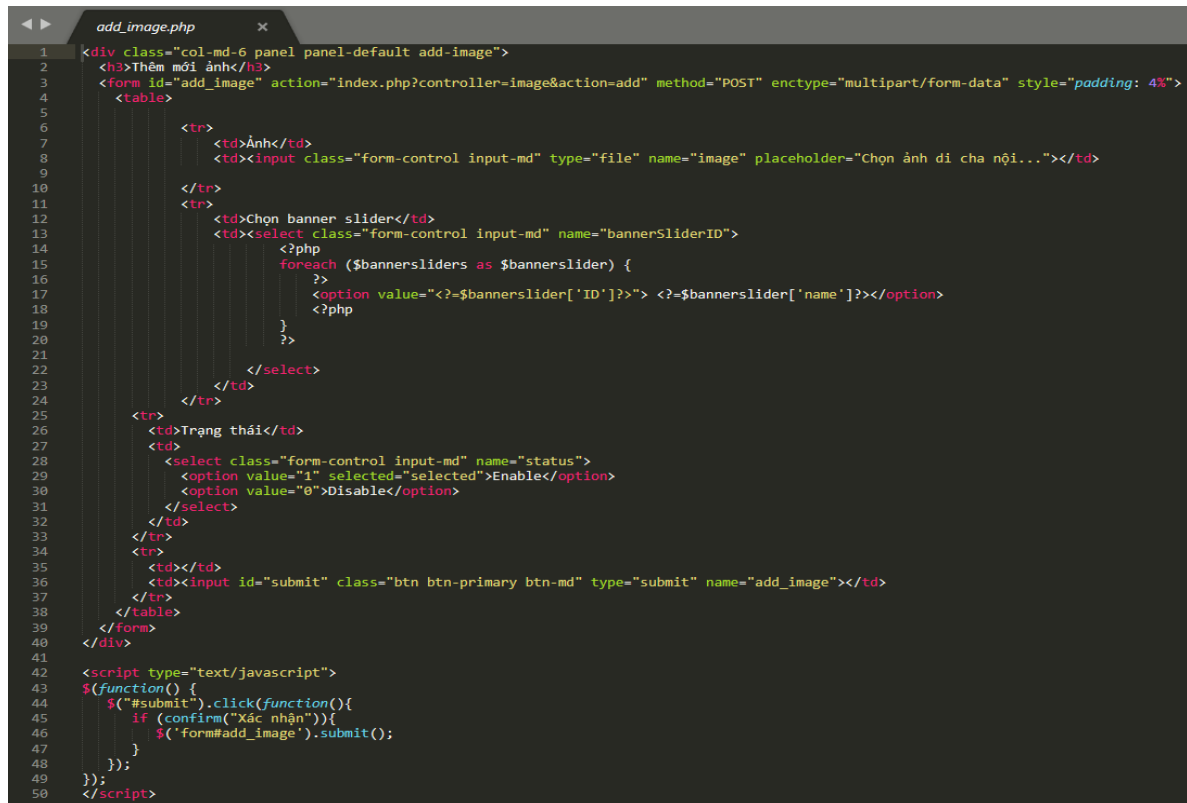
c. Chức năng thêm ảnh cho Banner Slider

Để có thể thêm ảnh cho Banner Slider, ID của Banner Slider sẽ được gửi lên bằng phương thức GET nếu người dùng bấm vào biểu tượng bút chì ở cột danh sách ảnh:

LOGO	
User picture User Account Status	
Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out	Thêm mới ảnh Ảnh Chọn tệp Chọn Banner Slider Homepage Trạng thái Enable Xác nhận

Hình 2.43: Thiết kế giao diện thêm ảnh cho Banner Slider

Để có thể hiển thị được biểu mẫu thêm ảnh mới, view **add_image.php** được tạo ra:



```
1 <div class="col-md-6 panel panel-default add-image">
2 <h3>Thêm mới ảnh</h3>
3 <form id="add_image" action="index.php?controller=image&action=add" method="POST" enctype="multipart/form-data" style="padding: 4px">
4 <table>
5
6 <tr>
7 <td>Ảnh</td>
8 <td><input class="form-control input-md" type="file" name="image" placeholder="Chọn ảnh di cha nội..."></td>
9
10 </tr>
11 <tr>
12 <td>Chọn banner slider</td>
13 <td><select class="form-control input-md" name="bannerSliderID">
14 <?php
15 <foreach ($bannersliders as $bannerSlider) {
16 <?
17 <option value="<?-$bannerSlider['ID']">"> <?-$bannerSlider['name']></option>
18 <?php
19 <?
20 <?
21 </foreach>
22 </select>
23 </td>
24 </tr>
25 <tr>
26 <td>Trạng thái</td>
27 <td>
28 <select class="form-control input-md" name="status">
29 <option value="1" selected="selected">Enable</option>
30 <option value="0">Disable</option>
31 </select>
32 </td>
33 </tr>
34 <tr>
35 <td></td>
36 <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="add_image"></td>
37 </tr>
38 </table>
39 </form>
40 </div>
41
42 <script type="text/javascript">
43 $(function() {
44 $("#submit").click(function(){
45 if (confirm("Xác nhận")){
46 $('form#add_image').submit();
47 }
48 });
49 });
50 </script>
```

Hình 2.44: Mã nguồn chức năng thêm mới ảnh cho Banner Slider

Trong view **add_image.php**, form **add_image** sẽ được dùng để gửi các dữ liệu trong các trường bằng phương thức POST.

Controller **index.php** trong thư mục Admin/controller/image/ được dùng để nhận và xử lý các yêu cầu dữ liệu được lên liên quan tới các chức năng quản lý ảnh.

Trong trường hợp thêm ảnh mới vào trong một Banner Slider, hành động ‘**add**’ sẽ được thực thi và tương tự như các chức năng thêm mới item đã nêu ở 2 phần trước, câu lệnh **InsertData()** sẽ được gọi để thêm mới tên ảnh vào trong database.

Tương tự, hàm **move_upload_file()** cũng được gọi để thêm mới ảnh vào trong thư mục pub/image/.

```

1 <?php
2 include "model/image.php";
3 $db = new Image;
4 $db->connect();
5 $tblTable = 'image';
6 $subTable1 = 'bannerslider';
7 $bannersliders = $db->getAllData($subTable1);
8 if (isset($_GET['action'])) {
9     $action = $_GET['action'];
10 } else {
11     $action = '';
12 }
13 switch ($action) {
14     case 'add':{
15         if (isset($_POST['add_image'])) {
16             $bannerSliderID = $_POST['bannerSliderID'];
17             $source = $_FILES['image']['name'];
18             $status = $_POST['status'];
19             $db->InsertData($bannerSliderID, $source, $status);
20             if (isset($_FILES['image'])) {
21                 move_uploaded_file($_FILES['image']['tmp_name'], 'pub/image/' . $_FILES['image']['name']);
22             }
23             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=image&action=list';</script>");
24         }
25         require_once('view/image/add_image.php');
26         break;
27     }
28     case 'list':{
29         if ($db->getAllData($tblTable)) {
30             $data = $db->getAllData($tblTable);
31         } else $data = null;
32         require_once('view/image/list_image.php');
33         break;
34     }
35     case 'delete':{
36         $id = $_GET['ID'];
37         $image = $_GET['image'];
38         unlink('pub/image/' . $image);
39         $db->DeleteDataByID($tblTable, $id);
40         $data = $db->getAllData($tblTable);
41         require_once('view/image/list_image.php');
42         break;
43     }
44 }
45 }
46 }
47

```

Hình 2.45: Mã nguồn giao diện thêm mới ảnh cho Banner Slider

3.3. Các chức năng quản lý bài viết

a. Chức năng thêm mới bài viết

LOGO	
User picture User Account Status Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out	Thêm mới bài viết Tên bài viết Tên bài viết... Mô tả ngắn Mô tả ngắn... Nội dung Nội dung Chọn banner Homepage Trạng thái Enable Xác nhận

Hình 2.46: Thiết kế giao diện thêm mới bài viết

Vì bài viết có nhiều dữ liệu cần được nhập hơn các mục khác nên biểu mẫu của phần này sẽ được làm to hơn. Trường nội dung không còn được tạo bằng thẻ `input` nữa mà ta sẽ dùng thẻ `<textarea>`.

```
add_post.php
1 <div class="col-md-10 panel panel-default add-post">
2   <h3>Thêm mới bài viết</h3>
3   <form id="add_post" action="index.php?controller=post&action=add" method="POST" enctype="multipart/form-data" style="padding: 4%">
4     <table style="width: 100%;">
5       <tr>
6         <td>Tên bài viết</td>
7         <td><input class="form-control input-md" type="text" name="name" placeholder="Tên bài viết..."></td>
8       </tr>
9
10      <tr>
11        <td>Mô tả ngắn</td>
12        <td><textarea class="form-control input-md" name="shortDescription" rows="2" cols="30" placeholder="Mô tả ngắn..."></textarea></td>
13      </tr>
14
15      <tr>
16        <td>Nội dung</td>
17        <td><textarea class="form-control input-md" name="content" rows="11" cols="30" placeholder="Nội dung..."></textarea></td>
18      </tr>
19      <tr>
20        <td>Chọn banner</td>
21        <td><select class="form-control input-md" name="bannerID">
22          <?php
23            foreach ($banners as $banner) {
24              <?
25                <option value="<?=$banner['ID']?>"><?=$banner['name']?></option>
26              <?php
27            }
28          <?php
29        </select>
30      </td>
31    </tr>
32    <tr>
33      <td>Trạng thái</td>
34      <td><select class="form-control input-md" name="status">
35        <option value="1" selected="selected">Enable</option>
36        <option value="0">Disable</option>
37      </select>
38    </td>
39  </tr>
40  <tr>
41    <td></td>
42    <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="add_post"></td>
43  </tr>
44 </table>
45 </form>
46 </div>
```

Hình 2.47: Mã nguồn giao diện thêm mới bài viết

Form ‘**add_post**’ sẽ gửi những thông tin về bài viết mà người dùng nhập vào các trường tới controller bằng phương thức POST.

Danh sách Banner Slider cũng được gọi ra để người dùng có thể chọn Banner Slider hiển thị cho bài viết đó.

Controller sẽ lấy các dữ liệu được gửi tới, sau đó gọi và truyền vào hàm **InsertData()** để nhập bài viết mới vào trong database.

```

14 switch ($action) {
15     case 'list':{
16         require_once('view/post/list_post.php');
17         break;
18     }
19
20     case 'add':{
21         if (isset($_POST['add_post'])) {
22             $name = $_POST['name'];
23             $shortDescription = $_POST['shortDescription'];
24             $content = $_POST['content'];
25             $bannerID = $_POST['bannerID'];
26             $status = $_POST['status'];
27             $db->InsertData($name, $shortDescription, $content, $bannerID, $status);
28
29             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=post&action=list';</script>");
30         }
31         require_once('view/post/add_post.php');
32         break;
33     }
34
35     case 'edit':{
36         if (isset($_GET['ID'])) {
37             $id = $_GET['ID'];
38             $postByID = $db->getDataByID($tblTable,$id);
39         }
40
41         require_once('view/post/edit_post.php');
42         break;
43     }
44
45     case 'update_post':{
46         if (isset($_POST['update_post'])) {
47             $id = $_GET['ID'];
48             $name = $_POST['name'];
49             $shortDescription = $_POST['shortDescription'];
50             $content = $_POST['content'];
51             $bannerID = $_POST['bannerID'];
52             $status = $_POST['status'];
53             $db->UpdateData($id, $name, $shortDescription, $content, $bannerID, $status);
54
55             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=post&action=list';</script>");
56         }
57         break;
58     }
59 }
60
61
62 case 'delete':{
63     $id = $_GET['ID'];
64     $db->DeleteDataByID($tblTable , $id);
65     $data = $db->getAllData($tblTable);
66     echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=post&action=list';</script>");

```

Hình 2.48: Mã nguồn chức năng thêm mới bài viết

Hàm InsertData() trong Admin/model/post:

```

1 <?php
2 include "DBConnect.php";
3 /**
4  *
5  */
6 class Post extends Database
7 {
8     //them du lieu
9     public function InsertData($name, $shortDescription, $content, $bannerID, $status){
10         $sql = "INSERT INTO post(name, shortDescription, content, bannerID, status)
11             VALUES ('$name', '$shortDescription', '$content', '$bannerID', '$status')";
12         return $this->execute($sql);
13     }
14
15     //sua du lieu
16     public function UpdateData($id, $name, $shortDescription, $content, $bannerID, $status){
17         $sql = "UPDATE post SET name='$name', shortDescription='$shortDescription', content='$content', bannerID='$bannerID', status='$status' WHERE ID = '$id' ";
18         return $this->execute($sql);
19     }
20 }
21
22
23 >>

```

Hình 2.49: Mã nguồn chức năng thêm mới bài viết vào cơ sở dữ liệu

b. Chức năng hiển thị danh sách bài viết

The wireframe shows a web application layout. At the top is a header bar with a 'LOGO' placeholder on the left. Below the header is a sidebar on the left containing a 'User Account' section with a 'User picture' and 'Status' labels. Below this is a list of menu items: 'Danh sách đồ uống', 'Danh sách thể loại đồ uống', 'Danh sách Banner', 'Danh sách ảnh', 'Danh sách bài viết', 'Danh sách tài khoản', and 'Log out'. The main content area on the right is titled 'Danh sách bài viết' and contains a table with the following columns: 'ID', 'Tên bài viết', 'Mô tả ngắn', 'Nội dung', 'Trạng thái', and 'Hoạt động'. The table has four rows, each with a pencil icon and a trash can icon in the 'Hoạt động' column. Below the table is a button labeled 'Thêm mới bài viết'.

Hình 2.50: Thiết kế giao diện hiển thị danh sách bài viết

Hàm **getAllData('tableName')** được dùng để thực thi câu lệnh sql: **SELECT * FROM tableName** để lấy tất cả các dữ liệu trong bảng **'tableName'**.

```
case 'list':{
    if ($db->getAllData($tblTable)) {
        $data = $db->getAllData($tblTable);
    } else $data = null;

    require_once('view/bannerslider/list_bannerslider.php');
    break;
}
```

Hình 2.51: Mã nguồn chức năng hiển thị danh sách bài viết

Vòng lặp foreach sẽ được dùng để in tất cả các bài viết có trong database ra màn hình:

```
foreach ($data as $value) {
    <tr style="border-bottom: 1px solid black;">
        <td><?= $value['ID'] ?></td>
        <td><?= $value['name'] ?></td>
        <td>
            <?php
                foreach ($images as $image) {
                    if ($image['bannerID'] == $value['ID'] ){
                        
                    }
                }
            <?php
            <ul class="panel-settings panel-button-tab-right" style="float:right;">
                <li>
                    <a class="btn-link pull-right" href="http://localhost.com/helentea/admin/index.php?controller=image&action=list&bannersliderID=<?= $value['ID'] ?>">
```

Hình 2.52: Mã nguồn chức năng hiển thị danh sách bài viết

c. Chức năng xóa bài viết

Danh sách bài viết					
ID	Tên bài viết	Mô tả ngắn	Nội dung	Trạng thái	Hoạt động
					 
					 

Hình 2.53: Thiết kế giao diện chức năng xóa bài viết

Khi người dùng bấm vào biểu tượng thùng rác, yêu cầu xóa và ID của bài viết tương ứng sẽ được gửi đi. Khi controller nhận được yêu cầu xóa từ người dùng và ID được lấy bằng phương thức **GET**, hàm **DeleteDataByID()** sẽ được gọi ra để xóa hàng trong bảng với ID tương ứng.

Ngoài ra, câu lệnh **unlink()** còn được dùng để xóa luôn ảnh trong thư mục.

Ảnh dưới đây là đoạn code trong controller phục vụ các tác vụ trên:

```
case 'delete':{
    $id = $_GET['ID'];
    $db->DeleteDataByID($tblTable , $id);
    $data = $db->getAllData($tblTable);
    echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=post&action=list';</script>");
    require_once('view/post/list_post.php');
    break;
}
```

Hình 2.54: Mã nguồn chức năng xóa bài viết

Đây là hàm **DeleteDataByID()** có chức năng gọi và thực thi câu lệnh sql nằm trong phần model:

```
//lay 1 row
public function DeleteDataByID($table, $id)
{
    $sql = "DELETE FROM $table WHERE ID = $id";
    return $this->execute($sql);
}
```

Hình 2.55: Mã nguồn chức năng xóa bài viết trong cơ sở dữ liệu

d. Chức năng sửa và cập nhật bài viết

Danh sách bài viết

ID	Tên bài viết	Mô tả ngắn	Nội dung	Trạng thái	Hoạt động
					 
					 

Hình 2.56: Thiết kế giao diện chức năng sửa và cập nhật bài viết

Khi người dùng bấm vào biểu tượng bút chì, trang cập nhật thông tin bài viết sẽ được chuyển tới với các thông tin và ID của hàng tương ứng

LOGO	
User picture User Account Status	Sửa bài viết Tên bài viết Bạn nên biết nguồn gốc cà phê ở đâu? Mô tả ngắn Chúng ta thường xuyên uống cà phê... Nội dung Chọn banner Homepage Trạng thái Enable Xác nhận
Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out	

Hình 2.57: Thiết kế giao diện sửa và cập nhật bài viết

Để thực hiện được các hành động đã nêu bên trên, đây là đoạn code trong controller sẽ được chạy:

```
case 'edit':{
    if (isset($_GET['ID'])) {
        $id = $_GET['ID'];
        $postByID = $db->getDataByID($tblTable,$id);
    }

    require_once('view/post/edit_post.php');
    break;
}
```

Hình 2.58: Mã nguồn chức năng sửa và cập nhật bài viết

Khi yêu cầu **'edit'** được gửi đi và có bao gồm ID, các thông tin của bài viết tương ứng sẽ được gọi ra bằng hàm **getDataByID()**.

Trong phần view, một form tương tự với phần thêm mới bài viết được tải nhưng những giá trị của các trường đã được gán sẵn tương ứng với dữ liệu của bài viết được

yêu cầu sửa. Form này cũng có các trường(field) tương tự như mục thêm mới bài viết để người dùng có thể dễ dàng chỉnh sửa mọi thông tin của bài viết.

```
edit_post.php
1 <div class="col-md-10 panel panel-default edit-post">
2 <h3>Sửa bài viết</h3>
3 <form id="update_post" action="index.php?controller=post&action=update_post&ID=?-$_POST['ID']" method="POST" enctype="multipart/form-data" style="padding: 40px">
4 <table style="width: 100%;">
5 <tr>
6 <td>Tên bài viết</td>
7 <td><input class="form-control input-md" type="text" value="<?=$_POST['name']?>" name="name" placeholder="Tên bài viết..."></td>
8 </tr>
9 <tr>
10 <td>Mô tả ngắn</td>
11 <td><textarea class="form-control input-md" name="shortDescription" rows="2" cols="30" placeholder="Mô tả ngắn..."><?=$_POST['shortDescription']?></textarea></td>
12 </tr>
13 <tr>
14 <td>Nội dung</td>
15 <td><textarea class="form-control input-md" name="content" rows="11" cols="30" placeholder="Nội dung..."><?=$_POST['content']?></textarea></td>
16 </tr>
17 <tr>
18 <td>Chọn banner</td>
19 <td><select class="form-control input-md" name="bannerID">
20 <?php
21 <foreach ($banners as $banner) {
22 <?php
23 <option value="<?=$_POST['ID']?>"><?=$_POST['ID']?></option>
24 <?php
25 <option value="<?=$_POST['ID']?>"><?=$_POST['ID']?></option>
26 <?php
27 <?php
28 <?php
29 <?php
30 </select>
31 </td>
32 </tr>
33 <tr>
34 <td>Trạng thái</td>
35 <td><select class="form-control input-md" name="status">
36 <option value="1" selected="selected">Enable</option>
37 <option value="0">Disable</option>
38 </select>
39 </td>
40 </tr>
41 <tr>
42 <td></td>
43 <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="update_post"></td>
44 </tr>
45 </table>
46 </form>
47 </div>
```

Hình 2.59: Mã nguồn giao diện sửa và cập nhật bài viết

Sau khi chỉnh sửa xong, nếu người dùng bấm vào nút gửi trong form edit, yêu cầu ‘**update_post**’ sẽ được gửi đi. Nếu người dùng có thay đổi ảnh, ảnh cũ của bài viết trong thư mục pub/image sẽ bị xóa và thay vào đó là ảnh mới.

```
case 'update_post':{
    if (isset($_POST['update_post'])) {
        $id = $_GET['ID'];
        $name = $_POST['name'];
        $shortDescription = $_POST['shortDescription'];
        $content = $_POST['content'];
        $bannerID = $_POST['bannerID'];
        $status = $_POST['status'];
        $db->updateData($id, $name, $shortDescription, $content, $bannerID, $status);
        echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=post&action=list';</script>");
    }
    break;
}
```

Hình 2.60: Mã nguồn chức năng sửa và cập nhật bài viết

3.4. Các chức năng quản lý tài khoản

a. Chức năng thêm mới tài khoản

LOGO		
User Account Status	Thêm mới tài khoản	
Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out	Username	
	Password	
	Phone Number	
	Chức vụ Admin	
	Xác nhận	

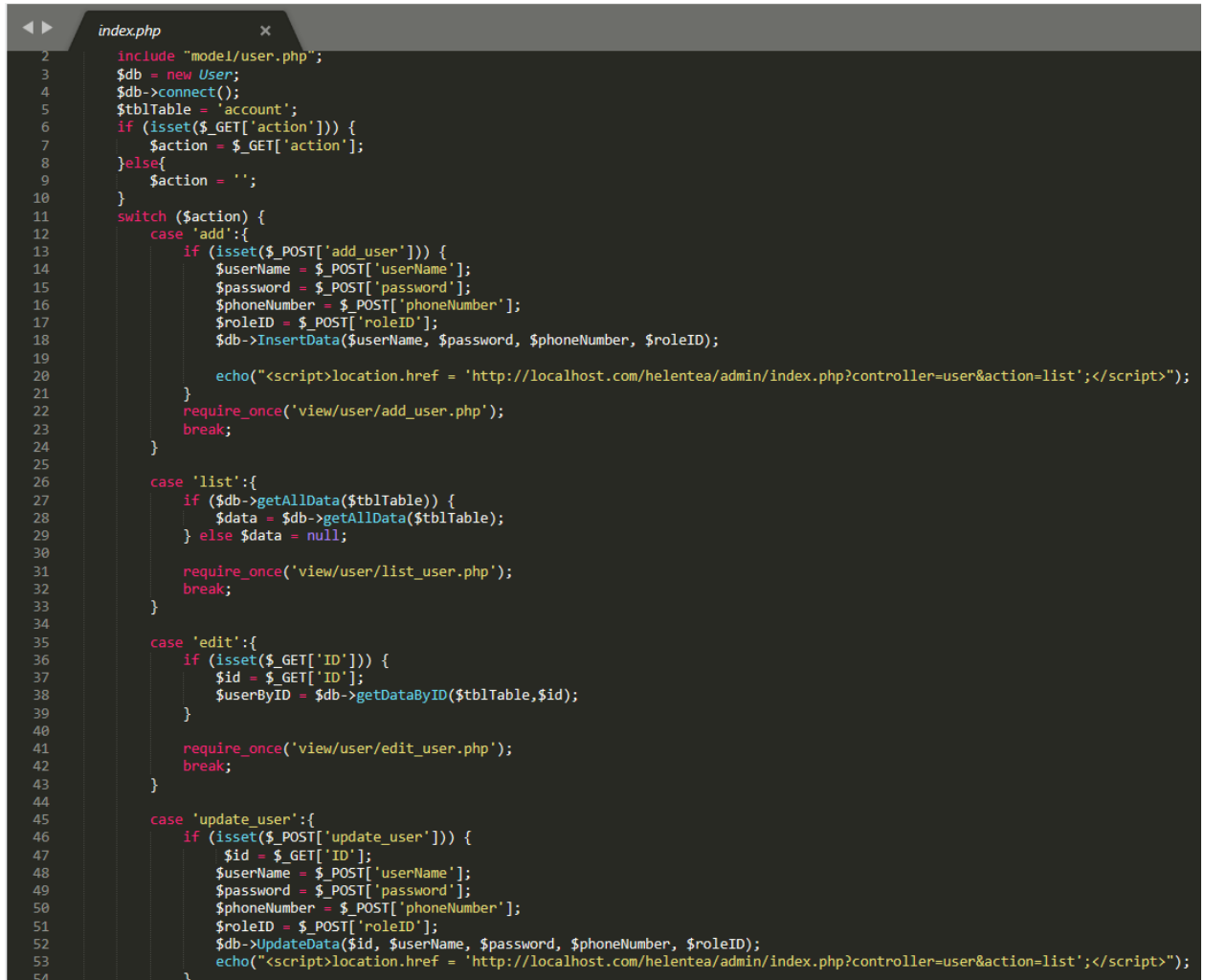
Hình 2.61: Thiết kế giao diện chức năng thêm mới tài khoản

```
add_user.php
1 <div class="col-md-6 panel panel-default add-user">
2 <h3>Thêm mới tài khoản</h3>
3 <form id="add_user" action="index.php?controller=user&action=add" method="POST" enctype="multipart/form-data" style="padding: 4%">
4 <table>
5 <tr>
6 <td>Username</td>
7 <td><input class="form-control input-md" type="text" name="userName"></td>
8 </tr>
9 <tr>
10 <td>Password</td>
11 <td><input class="form-control input-md" type="password" name="password"></td>
12 </tr>
13 <tr>
14 <td>Phonenumber</td>
15 <td><input class="form-control input-md" type="text" name="phoneNumber"></td>
16 </tr>
17 <tr>
18 <td>Chức vụ</td>
19 <td><select class="form-control input-md" name="roleID">
20 <option value="1">Admin</option>
21 <option value="2">Staff</option>
22 </select>
23 </td>
24 </tr>
25 <tr>
26 <td></td>
27 <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="add_user"></td>
28 </tr>
29 </table>
30 </form>
31 </div>
32
33 <script type="text/javascript">
34 $(function() {
35 <#submit>.click(function(){
36 if (confirm("Xác nhận")){
37 $('form#add_user').submit();
38 }
39 });
40 });
41 </script>
```

Hình 2.62: Mã nguồn giao diện thêm mới tài khoản

Form ‘add_user’ sẽ gửi những thông tin về tài khoản mà người dùng nhập vào các trường tới controller bằng phương thức POST.

Danh sách vị trí cũng được gọi ra để người dùng có thể phân quyền cho tài khoản.



```
2 include "model/user.php";
3 $db = new User;
4 $db->connect();
5 $tblTable = 'account';
6 if (isset($_GET['action'])) {
7     $action = $_GET['action'];
8 }else{
9     $action = '';
10 }
11 switch ($action) {
12     case 'add':{
13         if (isset($_POST['add_user'])) {
14             $userName = $_POST['userName'];
15             $password = $_POST['password'];
16             $phoneNumber = $_POST['phoneNumber'];
17             $roleID = $_POST['roleID'];
18             $db->InsertData($userName, $password, $phoneNumber, $roleID);
19
20             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=user&action=list';</script>");
21         }
22         require_once('view/user/add_user.php');
23         break;
24     }
25
26     case 'list':{
27         if ($db->getAllData($tblTable)) {
28             $data = $db->getAllData($tblTable);
29         } else $data = null;
30
31         require_once('view/user/list_user.php');
32         break;
33     }
34
35     case 'edit':{
36         if (isset($_GET['ID'])) {
37             $id = $_GET['ID'];
38             $userByID = $db->getDataByID($tblTable,$id);
39         }
40
41         require_once('view/user/edit_user.php');
42         break;
43     }
44
45     case 'update_user':{
46         if (isset($_POST['update_user'])) {
47             $id = $_GET['ID'];
48             $userName = $_POST['userName'];
49             $password = $_POST['password'];
50             $phoneNumber = $_POST['phoneNumber'];
51             $roleID = $_POST['roleID'];
52             $db->UpdateData($id, $userName, $password, $phoneNumber, $roleID);
53             echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=user&action=list';</script>");
54         }
55     }
```

Hình 2.63: Mã nguồn chức năng thêm mới tài khoản

Hàm InsertData() trong Admin/model/user:

```
public function InsertData($userName, $password, $phoneNumber, $roleID){
    $sql = "INSERT INTO account(userName, password, phoneNumber, roleID)
    VALUES ('$userName', '$password', '$phoneNumber', '$roleID')";
    return $this->execute($sql);
}
```

Hình 2.64: Mã nguồn chức năng thêm mới tài khoản vào cơ sở dữ liệu

b. Chức năng hiển thị danh sách tài khoản

ID	Uername	Password	Phone Number	Role	Hoạt động

Thêm tài khoản mới

Hình 2.65: Thiết kế giao diện hiển thị danh sách tài khoản

Hàm **getAllData('tableName')** được dùng để thực thi câu lệnh sql: **SELECT * FROM tableName** để lấy tất cả các dữ liệu trong bảng **'tableName'**.

```
case 'list':{
    if ($db->getAllData($tblTable)) {
        $data = $db->getAllData($tblTable);
    } else $data = null;

    require_once('view/bannerslider/list_bannerslider.php');
    break;
}
```

Hình 2.66: Mã nguồn chức năng hiển thị danh sách tài khoản

Vòng lặp foreach sẽ được dùng để in tất cả các tài khoản có trong database ra màn hình:

```
<?php
if (isset($data)) {
    foreach ($data as $value) {
        ?>

        <tr style="border-bottom: 1px solid black;">
            <td><?= $value['ID'] ?></td>
            <td><?= $value['userName'] ?></td>
            <td><?= $value['password'] ?></td>
            <td><?= $value['phoneNumber'] ?></td>
            <td>
                <?php
                if ($value['roleID'] == 1) {
                    echo "Admin";
                }
                if ($value['roleID'] == 2) {
                    echo "Staff";
                }
                ?>
            </td>
            <td>
                <div>
```

Hình 2.66: Mã nguồn chức năng hiển thị danh sách tài khoản

c. Chức năng xóa tài khoản

Danh sách tài khoản					
ID	Username	Password	Phone Number	Role	Hoạt động
					 
					 

Hình 2.67: Thiết kế giao diện chức năng xóa tài khoản

Khi người dùng bấm vào biểu tượng thùng rác, yêu cầu xóa và ID của tài khoản tương ứng sẽ được gửi đi. Khi controller nhận được yêu cầu xóa từ người dùng và ID được lấy bằng phương thức **GET**, hàm **DeleteDataByID()** sẽ được gọi ra để xóa hàng trong bảng với ID tương ứng.

Ngoài ra, câu lệnh **unlink()** còn được dùng để xóa luôn ảnh trong thư mục.

Ảnh dưới đây là đoạn code trong controller phục vụ các tác vụ trên:

```
case 'delete':{
    $id = $_GET['ID'];
    $db->DeleteDataByID($tblTable , $id);
    $data = $db->getAllData($tblTable);
    require_once('view/user/list_user.php');
    break;
}
```

Hình 2.68: Mã nguồn chức năng xóa tài khoản

Đây là hàm **DeleteDataByID()** có chức năng gọi và thực thi câu lệnh sql nằm trong phần model:

```
//lay 1 row
public function DeleteDataByID($table, $id)
{
    $sql = "DELETE FROM $table WHERE ID = $id";
    return $this->execute($sql);
}
```

Hình 2.69: Mã nguồn chức năng xóa tài khoản trong cơ sở dữ liệu

d. Chức năng sửa và cập nhật tài khoản

Danh sách tài khoản

ID	Username	Password	Phone Number	Role	Hoạt động
					 
					 

Hình 2.70: Thiết kế giao diện chức năng sửa và cập nhật tài khoản

Khi người dùng bấm vào biểu tượng bút chì, trang cập nhật thông tin tài khoản sẽ được chuyển tới với các thông tin và ID của hàng tương ứng.

LOGO		
User picture	User Account Status	Sửa tài khoản
Danh sách đồ uống Danh sách thể loại đồ uống Danh sách Banner Danh sách ảnh Danh sách bài viết Danh sách tài khoản Log out		Username: staff1
		Password: 123
		Phone Number: 0964481482
		Chức vụ: Staff
		Xác nhận

Hình 2.71: Thiết kế giao diện sửa và cập nhật tài khoản

Để thực hiện được các hành động đã nêu bên trên, đây là đoạn code trong controller sẽ được chạy:

```
case 'edit':{
    if (isset($_GET['ID'])) {
        $id = $_GET['ID'];
        $userByID = $db->getDataByID($tblTable,$id);
    }

    require_once('view/user/edit_user.php');
    break;
}
```

Hình 2.72: Mã nguồn chức năng sửa và cập nhật tài khoản

Khi yêu cầu **'edit'** được gửi đi và có bao gồm ID, các thông tin của tài khoản tương ứng sẽ được gọi ra bằng hàm **getDataByID()**.

Trong phần view, một form tương tự với phần thêm mới tài khoản được tải nhưng những giá trị của các trường đã được gán sẵn tương ứng với dữ liệu của tài

khóa được yêu cầu sửa. Form này cũng có các trường(field) tương tự như mục thêm mới tài khoản để người dùng có thể dễ dàng chỉnh sửa mọi thông tin của tài khoản.

```
edit_user.php
1 <div class="col-md-6 panel panel-default add-user">
2   <h3>Sửa tài khoản</h3>
3   <form id="submit" action="index.php?controller=user&action=update_user&ID=?=$userByID['ID']" method="POST" enctype="multipart/form-data" style="padding: 4%
4   ">
5     <table>
6       <tr>
7         <td>Username</td>
8         <td><input class="form-control input-md" type="text" name="userName" value="<?=$userByID['userName']?>"></td>
9       </tr>
10      <tr>
11        <td>Password</td>
12        <td><input class="form-control input-md" type="text" name="password" value="<?=$userByID['password']?>"></td>
13      </tr>
14      <tr>
15        <td>Phonenumber</td>
16        <td><input class="form-control input-md" type="text" name="phoneNumber" value="<?=$userByID['phoneNumber']?>"></td>
17      </tr>
18      <tr>
19        <td>Chức vụ</td>
20        <td><select class="form-control input-md" name="roleID">
21          <?php
22            if ($userByID['roleID'] == 1) {
23              <option value="1" selected="selected">Admin</option>
24              <option value="2">Staff</option>
25            }else{
26              <option value="1">Admin</option>
27              <option value="2" selected="selected">Staff</option><?php
28            }
29          </?php>
30        </select>
31      </td>
32    </tr>
33  </table>
34  <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="update_user"></td>
35  </tr>
36  </table>
37  </form>
38  </div>
39  <script type="text/javascript">
40    $(function() {
41      $("input#submit").click(function(){
42        if (confirm("Xác nhận")){
43          $('form#update_user').submit();
44        }
45      });
46    });
47  </script>
```

Hình 2.73: Mã nguồn giao diện sửa và cập nhật tài khoản

Sau khi chỉnh sửa xong, nếu người dùng bấm vào nút gửi trong form edit, yêu cầu ‘**update_user**’ sẽ được gửi đi

```
case 'update_user':{
  if (isset($_POST['update_user'])) {
    $id = $_GET['ID'];
    $userName = $_POST['userName'];
    $password = $_POST['password'];
    $phoneNumber = $_POST['phoneNumber'];
    $roleID = $_POST['roleID'];
    $db->UpdateData($id, $userName, $password, $phoneNumber, $roleID);
    echo("<script>location.href = 'http://localhost.com/helentea/admin/index.php?controller=user&action=list';</script>");
  }
  break;
}
```

Hình 2.74: Mã nguồn chức năng sửa và cập nhật tài khoản

3.5. Các chức năng đăng nhập và phân quyền

a. Chức năng đăng nhập

- Session:

Trong chức năng này ta sẽ dùng SESSION, là thông tin về phiên làm việc cho từng khách truy cập trong PHP nó tạo một file trong thư mục tạm (thư mục này cài đặt ở php.ini : session.save_path) để lưu thông tin này, thông tin này được dùng chung cho tất cả các trang mà khách truy cập. Mỗi khách truy cập, dựa vào thông tin trình duyệt gửi lên nó tạo (hoặc phục hồi) một Session riêng cho khách đó. Để hiểu rõ hơn ta sẽ bắt tay vào làm việc với SESSION.

Session trong PHP bắt đầu hoạt động sau khi gọi hàm **session_start()**.

Sau khi session được khởi động, ta có thể dùng biến toàn cục **\$_SESSION**, là nơi SESSION lưu trữ thông tin - cũng chính là dữ liệu phục hồi được giữa các trang, bạn truy cập biến này trong suốt vòng đời của session để lấy hoặc lưu trữ thông tin.

Với việc dùng session, ta có thể kiểm tra xem người dùng đã đăng nhập hay chưa hoặc hiển thị tên tài khoản trên nhiều trang khác nhau:

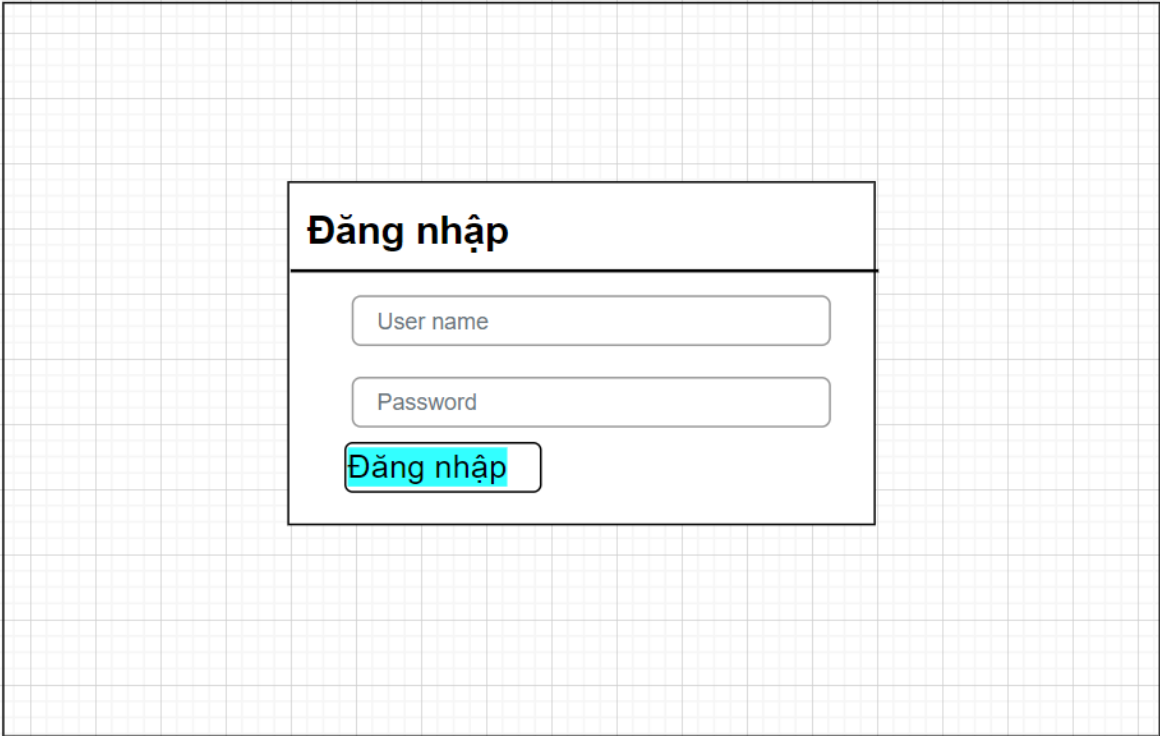
```
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

include "model/user.php";
$db = new User;
$db->connect();
$tblTable = 'account';
$users = $db->getAllData($tblTable);
if (isset($_POST['user_login'])) {

    $_SESSION['userName'] = $_POST['userName'];
    $_SESSION['password'] = $_POST['password'];
}
```

Hình 2.75: Mã nguồn chức năng kiểm tra đăng nhập và hiển thị tài khoản đăng nhập

Khi kiểm tra người dùng chưa đăng nhập, trang sẽ tự động chuyển về trang đăng nhập:



Đăng nhập

User name

Password

Đăng nhập

Hình 2.76: Thiết kế giao diện trang đăng nhập

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Helen Admin Login</title>
7 <link href="pub/css/bootstrap.min.css" rel="stylesheet">
8 <link href="pub/css/datepicker3.css" rel="stylesheet">
9 <link href="pub/css/styles.css" rel="stylesheet">
10 <!--[if lt IE 9]>
11 <script src="js/html5shiv.js"></script>
12 <script src="js/respond.min.js"></script>
13 <![endif]-->
14 </head>
15 <body>
16 <div class="row">
17 <div class="col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2 col-md-4 col-md-offset-4">
18 <div class="login-panel panel panel-default">
19 <div class="panel-heading">Đăng nhập</div>
20 <div class="panel-body">
21 <form id="user_login" action="index.php?controller=user&action=login" method="POST" enctype="multipart/form-data">
22 <table>
23 <tr>
24 <td>Tài khoản</td>
25 <td><input class="form-control input-md" type="text" name="userName" ></td>
26 </tr>
27 <tr>
28 <td>Mật khẩu</td>
29 <td><input class="form-control input-md" type="password" name="password"></td>
30 </tr>
31 <tr>
32 <td></td>
33 <td><input id="submit" class="btn btn-primary btn-md" type="submit" name="user_login"></td>
34 </tr>
35 </table>
36 </form>
37 </div>
38 </div>
39 </div><!-- /.col-->
40 </div><!-- /.row -->
41 <style>
42 .navbar,.portfolio-modal,#footerHelen{
43 display: none !important;
44 }
45 </style>
46
47 <script src="pub/js/jquery-1.11.1.min.js"></script>
48 <script src="pub/js/bootstrap.min.js"></script>
49
50 </body>
51 </html>

```

Hình 2.77: Mã nguồn giao diện trang đăng nhập

Người dùng có thể nhập tài khoản đã được tạo (ở chức năng tạo mới tài khoản), sau đó bấm nút gửi để gửi yêu cầu đăng nhập.

```

case 'login':{
    $users = $db->getAllData('account');
    $loggedIn = 0;
    $_SESSION['userName'] = $_POST['userName'];
    $_SESSION['password'] = $_POST['password'];
    var_dump('chết nè');
    foreach ($users as $user) {
        if (($user['userName'] == $_SESSION['userName']) && ($user['password'] == $_SESSION['password'])) {
            $loggedIn = 1;
        }
    }
    if ($loggedIn == 0) {
        echo("<script>location.href = 'http://localhost.com/helentea/admin/login.php';</script>");
    }
    break;
}
}

```

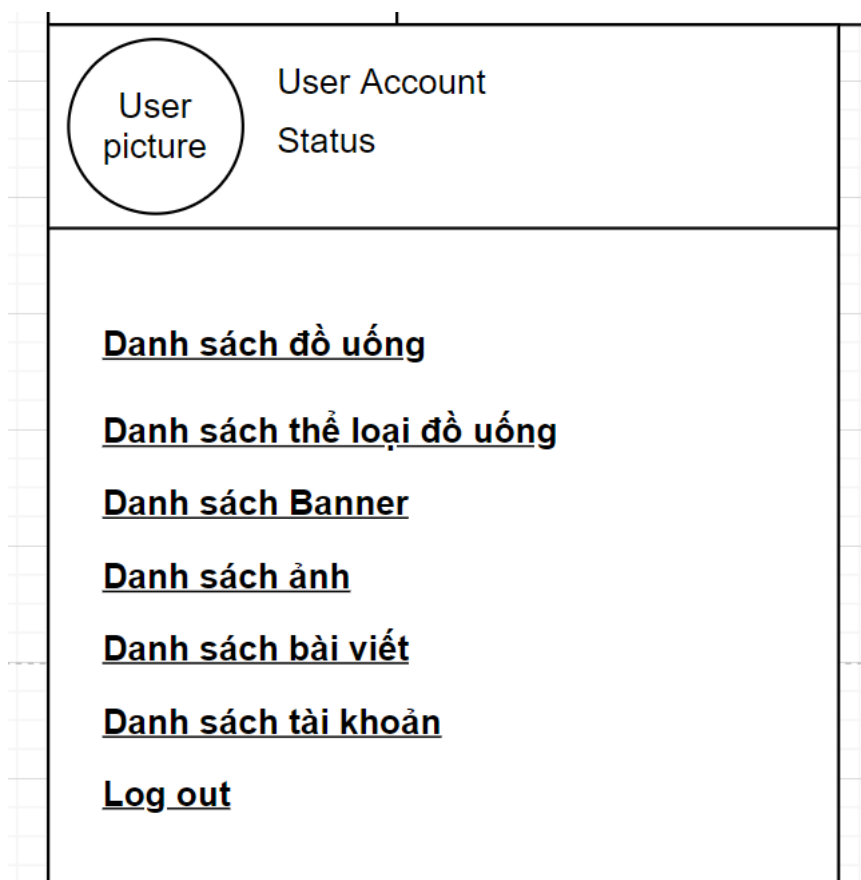
Hình 2.78: Mã nguồn chức năng đăng nhập

Khi nhận được yêu cầu đăng nhập từ phía người dùng, ta sẽ gán các biến toàn cục `$_SESSION['userName']` và `$_SESSION['password']` bằng tài khoản và mật

khẩu mà người dùng gửi lên, để người dùng sẽ không phải đăng nhập lại mỗi khi chuyển trang.

Tài khoản và mật khẩu được người dùng nhập sẽ được so sánh với database, nếu tài khoản và mật khẩu khớp thì sẽ trở về trang chủ hoặc sẽ quay lại trang đăng nhập.

Người dùng có thể đăng xuất bằng cách click vào nút **Log out** ở thanh điều hướng phía tay trái để đăng xuất.



Hình 2.79: Thiết kế giao diện chức năng đăng xuất

Nút này sẽ gửi yêu cầu **'logout'** tới controller sau đó controller sẽ thực thi xóa biến toàn cục `$_SESSION['password']`.

```
case 'logout':{
    $_SESSION['password'] = null;
    echo("<script>location.href = 'http://localhost.com/helentea/admin/login.php';</script>");
    break;
}
```

Hình 2.80: Mã nguồn chức năng đăng xuất

b. Chức năng phân quyền

Trong form thêm mới tài khoản, admin có thể chọn chức vụ cho tài khoản mới được tạo.

LOGO		
User Account Status	Sửa tài khoản	
User picture	Username	staff1
	Password	123
	Phone Number	0964481482
	Chức vụ	Admin Admin Staff
	Xác nhận	

Hình 2.81: Thiết kế giao diện chức năng phân quyền

Khi admin hoặc staff được chọn, form đăng ký này sẽ gửi lên biến **roleID** có giá trị là 1 hoặc 2 tương ứng.

```
<td>Chức vụ</td>  
<td><select class="form-control input-md" name="roleID">  
    <option value="1">Admin</option>  
    <option value="2">Staff</option>  
</select>  
</td>
```

Hình 2.82: Mã nguồn chức năng hiển thị phân quyền

Từ đó , mỗi tài khoản sẽ có thuộc tính **roleID** khác nhau tùy thuộc vào chức vụ được chọn.

Khi người dùng tiến hành đăng nhập, biến cục bộ **\$_COOKIE['roleID']** có giá trị bằng giá trị thuộc tính roleID của tài khoản đăng nhập sẽ được tạo.

Biến cục bộ **\$_COOKIE['roleID']** sẽ giúp ta kiểm tra được chức vụ của tài khoản đăng nhập và hạn chế hành động của tài khoản đó nếu có. Ví dụ khi người dùng yêu cầu hành động xem danh sách tài khoản hiện có trong cơ sở dữ liệu với **\$_COOKIE['roleID']** có giá trị là 2 tương ứng với chức vụ của tài khoản là nhân viên (staff), thì trang sẽ tự động trở về trang chủ thay vì lấy dữ liệu từ database và đổ ra view list_user.php.

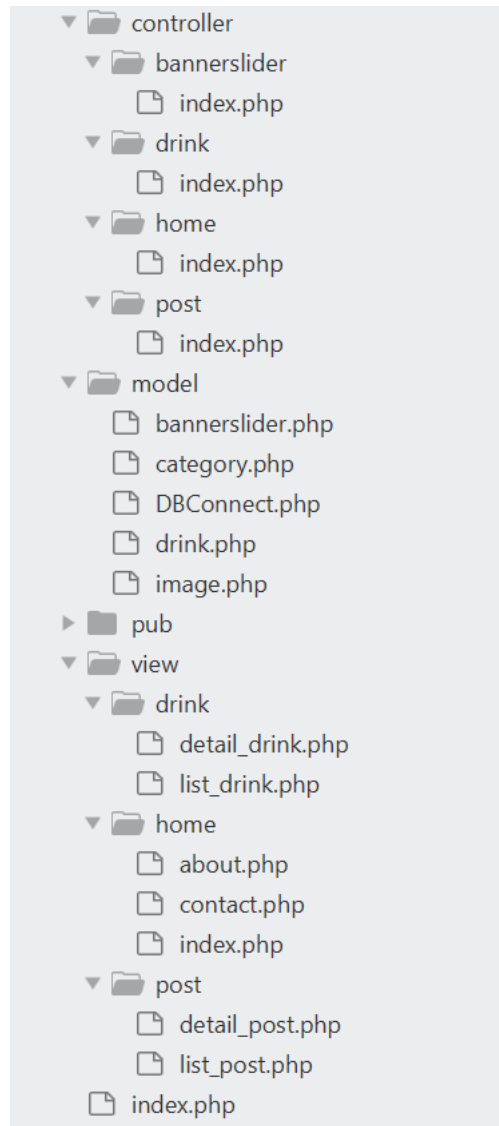
```
case 'list':{
    if ($_COOKIE['roleID'] == 1) {
        if ($db->getAllData($tblTable)) {
            $data = $db->getAllData($tblTable);
        } else $data = null;

        require_once('view/user/list_user.php');
        break;
    }else{
        echo("<script>location.href = 'http://localhost/helentea/admin/index.php';</script>");
    }
}
```

Hình 2.83: Mã nguồn chức năng phân quyền

4. Xây dựng Front-end

- Do website được xây dựng theo mô hình MVC nên mỗi chức năng sẽ đều có đủ các nhân tố Model, View và Controller.

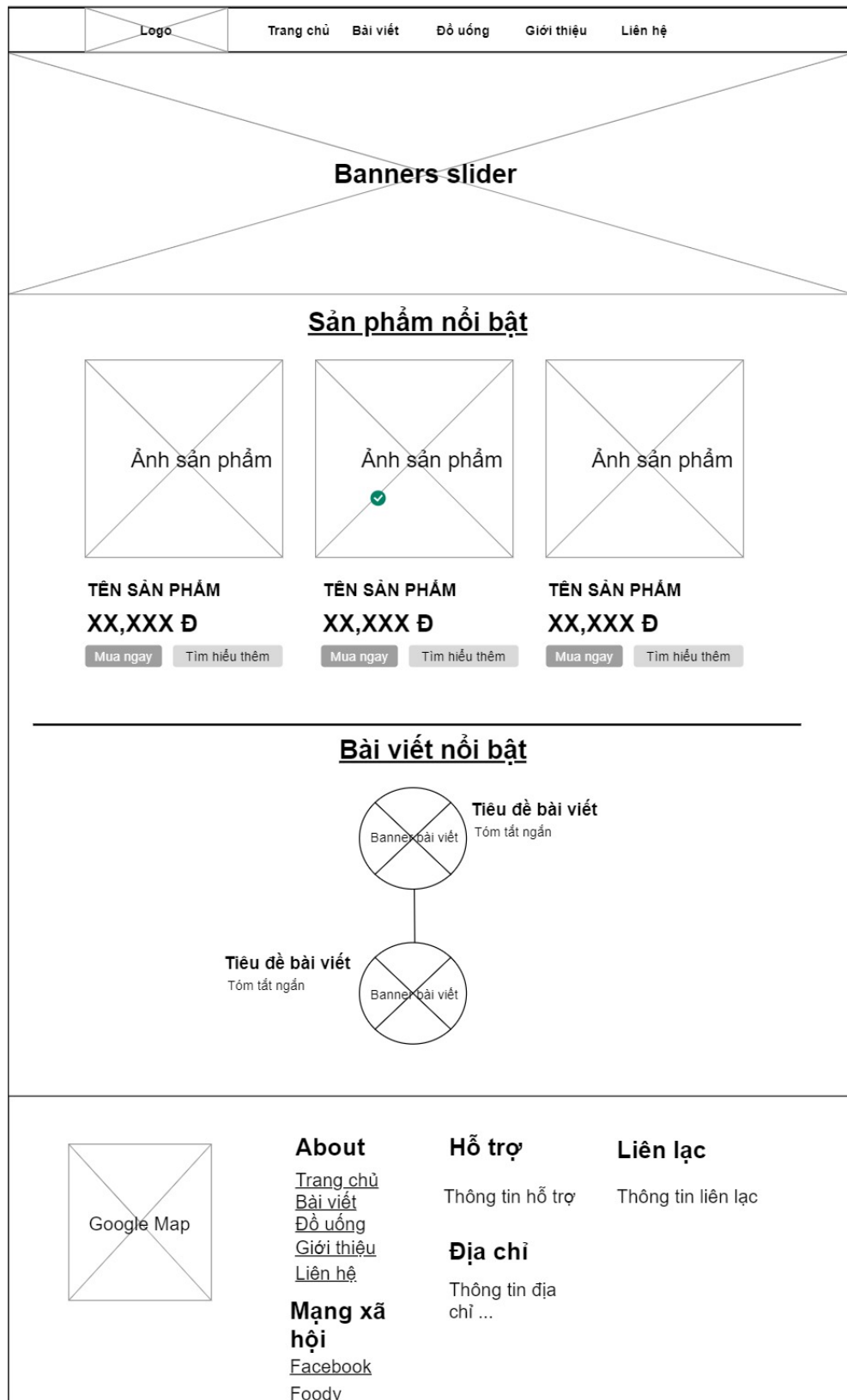


Hình 2.84: Cấu trúc chi tiết thư mục Front-end

- Tổng quan về Front-end:

- Website sử dụng HTML 5, được phát triển trên nền tảng ngôn ngữ HTML và quan trọng nhất của world wide web.
- Website sử dụng CSS làm ngôn ngữ tạo phong cách cho website.
- Website sử dụng framework Bootstrap 3, bao gồm HTML, CSS và JavaScript template để phát triển website chuẩn responsive.
- Website sử dụng ngôn ngữ lập trình Javascript và framework JQuery.

4.1. Trang chủ



Hình 2.85: Thiết kế giao diện trang chủ

Giao diện của trang chủ được tạo bằng file **index.php** trong thư mục **helentea/view/home**.

```
switch ($controller) {
    case 'drink':{
        echo '<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script><script src="view/drink/index.js"></script>';
        require_once('controller/drink/index.php');
        break;
    }

    case 'post':{
        echo '<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script><script src="view/drink/index.js"></script>';
        require_once('controller/post/index.php');
        break;
    }

    case 'home':{
        require_once('controller/home/index.php');
        break;
    }
    default:{
        require_once('controller/home/index.php');
    }
}
```

Hình 2.86: Mã nguồn chức năng hiển thị trang chủ

Khi người dùng không gửi bất kì yêu cầu nào ngoài trang chủ, controller **index.php** trong thư mục home sẽ được gọi.

```
<?php
include "model/drink.php";
$db = new Drink;
$db->connect();
$tblTable1 = 'drink';
$tblTable2 = 'category';
$tblTable3 = 'image';
$tblTable4 = 'post';
$posts = $db->getAllData($tblTable4);
$categories = $db->getAllData($tblTable2);
$drinks = $db->getAllData($tblTable1);
$images = $db->getAllData($tblTable3);

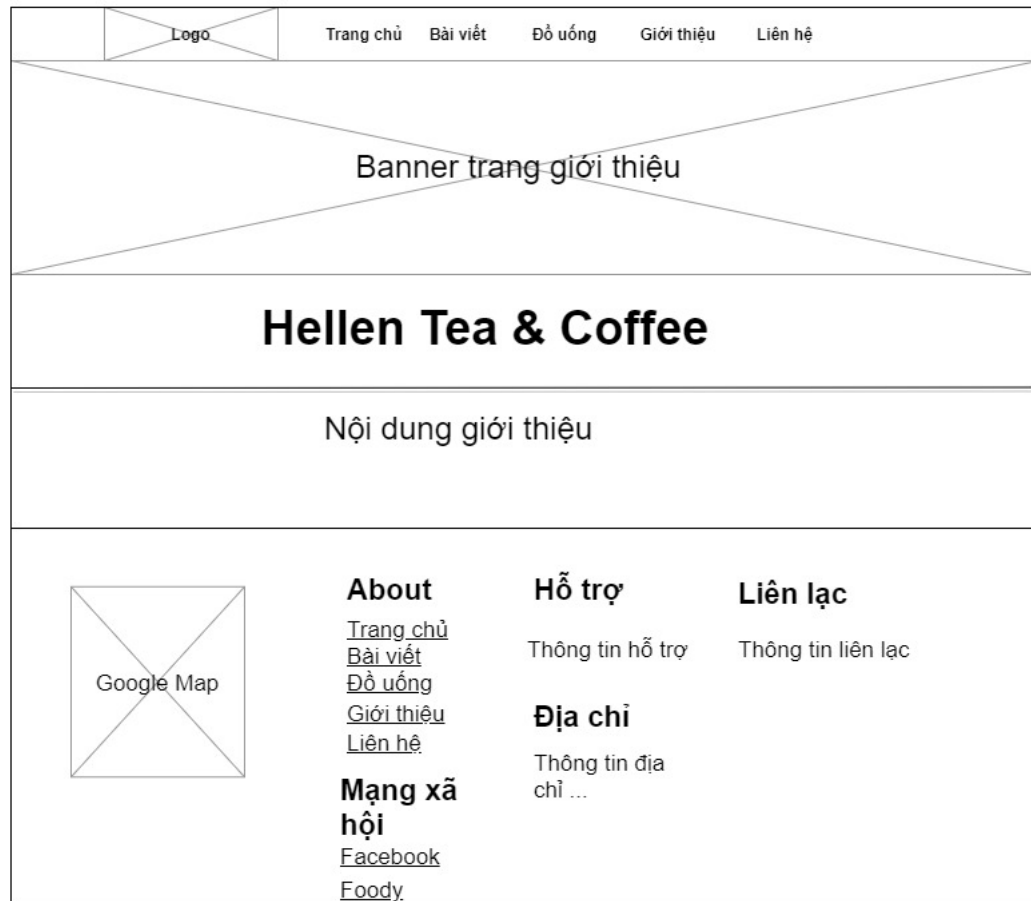
if (isset($_GET['action'])) {
    $action = $_GET['action'];
}else{
    $action = '';
}
if ($action) {
    switch ($action) {
        case 'about':{
            if (isset($_GET['categoryID'])){
                $categoryID = $_GET['categoryID'];
            }
            require_once('view/home/about.php');
            break;
        }
        case 'contact':{
            if (isset($_GET['id'])) {
                $id = $_GET['id'];
                $drinkByID = $db->getDataByID($tblTable,$id);
            }
            require_once('view/home/contact.php');
            break;
        }
    }
}

}else{
    require_once('view/home/index.php');
}
```

Hình 2.87: Mã nguồn giao diện trang chủ

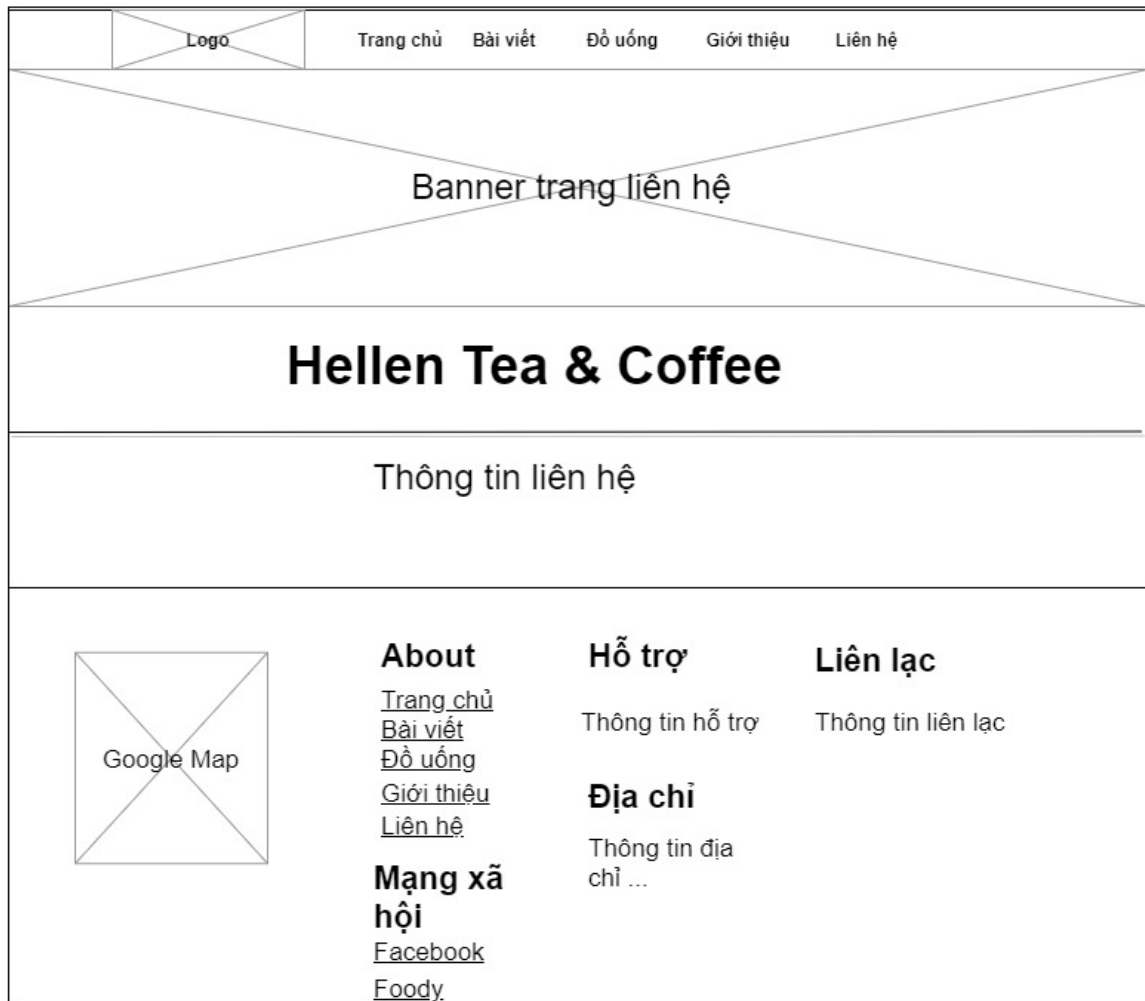
Ngoài ra người dùng có yêu cầu trang giới thiệu (about) hoặc trang liên hệ (contact), view **about.php** hoặc **contact.php** sẽ được gọi.

- *Trang giới thiệu:*



Hình 2.88: Thiết kế giao diện trang giới thiệu

- Trang liên hệ:



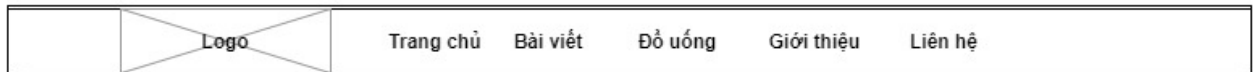
Hình 2.89: Thiết kế giao diện trang liên hệ

Dữ liệu trong các bảng sau cũng được lấy ra để hiển thị đa dạng trên trang chủ:

```
$tblTable1 = 'drink';  
$tblTable2 = 'category';  
$tblTable3 = 'image';  
$tblTable4 = 'post';
```

Hình 2.90: Mã nguồn để hiển thị các bảng

4.2. Thanh điều hướng



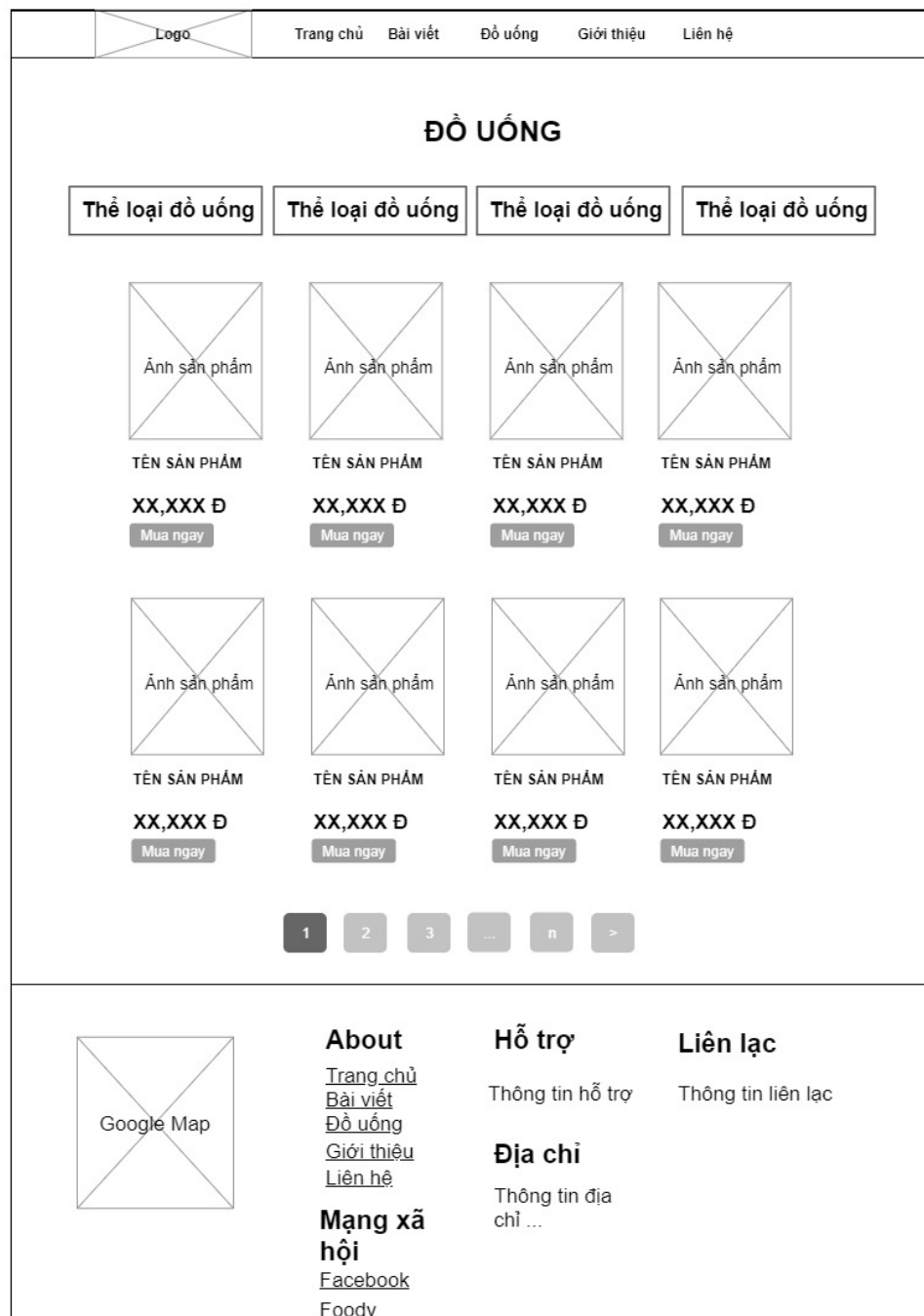
Hình 2.91: Thiết kế thanh điều hướng

```
<li class="nav-item"><a style="font-size: 18px;" class="nav-link js-scroll-trigger" href="index.php">Trang chủ</a></li>
<li class="nav-item"><a style="font-size: 18px;" class="nav-link js-scroll-trigger" href="index.php?controller=post&action=list">Bài viết</a></li>
<li class="nav-item"><a style="font-size: 18px;" class="nav-link js-scroll-trigger" href="index.php?controller=drink&action=list">Đồ uống</a></li>
<li class="nav-item"><a style="font-size: 18px;" class="nav-link js-scroll-trigger" href="index.php?controller=home&action=about">Giới thiệu</a></li>
<li class="nav-item"><a style="font-size: 18px;" class="nav-link js-scroll-trigger" href="index.php?controller=home&action=contact">Liên hệ</a></li>
```

Hình 2.92: Mã nguồn giao diện thanh điều hướng

Bằng các dùng các thẻ `<a></a>`, ta có thể gán các giá trị **href** nhằm gửi các yêu cầu và hành động lên các controller post, drink, home.

4.3. Trang danh sách đồ uống



Hình 2.93: Thiết kế giao diện trang danh sách đồ uống

Khi người dùng click vào ‘Đồ uống’ controller drink sẽ được gọi:

```
index.php
1 <?php
2 include "model/drink.php";
3 $db = new Drink;
4 $db->connect();
5 $tblTable = 'drink';
6 $subTable1 = 'category';
7 $subTable2 = 'image';
8 $categories = $db->getAllData($subTable1);
9 $drinks = $db->getAllData($tblTable);
10 $images = $db->getAllData($subTable2);
11 if (isset($_GET['action'])) {
12     $action = $_GET['action'];
13 }else{
14     $action = '';
15 }
16
17 switch ($action) {
18     case 'list':{
19         if (isset($_GET['categoryID'])){
20             $categoryID = isset($_GET['categoryID']);
21         }
22         require_once('view/drink/list_drink.php');
23         break;
24     }
25     case 'detail':{
26         if (isset($_GET['id'])) {
27             $id = $_GET['id'];
28             $drinkByID = $db->getDataByID($tblTable,$id);
29         }
30         require_once('view/drink/detail_drink.php');
31         break;
32     }
33 }
34
35 }
```

Hình 2.94: Mã nguồn hiển thị trang đồ uống

Nếu hành động được yêu cầu từ người dùng là ‘list’, ta sẽ kiểm tra xem tiếp người dùng có yêu cầu thể loại đồ uống hay không. Qua đó ta có thể hiển thị tất cả đồ uống hoặc một nhóm đồ uống thuộc cùng thể loại ra màn hình.

```
if (isset($_GET['categoryID'])){
    $categoryID = isset($_GET['categoryID']);
}
```

Hình 2.95: Mã nguồn hiển thị thể loại đồ uống

Khi đã lấy được dữ liệu trong các bảng bằng cách dùng hàm **getAllData()**, ta sẽ hiển thị chúng ra ngoài màn hình bằng cách dùng vòng lặp **foreach()** :

```
foreach ($drinks as $drink){
    if (isset($_POST['categoryID'])) {
        if ($drink['categoryID'] == $_POST['categoryID']) {
            >>
            <div class="col-lg-3 col-sm-6 mb-4" style="float:left;">
                <div class="portfolio-item">
                    <a class="portfolio-link" href="http://localhost.com/helentea/index.php?controller=drink&action=detail&id=?=$drink['ID']">>
                    </a>
                    <div class="portfolio-caption">
                        <div class="portfolio-caption-heading"><?= $drink['name']?></div>
                        <div class="portfolio-caption-subheading text-muted"><?= $drink['price']?> Đ</div>
                    </div>
                </div>
            </div>
            <?php
        }
    }
    else{
        >>
        <div class="col-lg-3 col-sm-6 mb-4" style="float:left;">
            <div class="portfolio-item">
                <a class="portfolio-link" href="http://localhost.com/helentea/index.php?controller=drink&action=detail&id=?=$drink['ID']">>
                </a>
                <div class="portfolio-caption">
                    <div class="portfolio-caption-heading"><?= $drink['name']?></div>
                    <div class="portfolio-caption-subheading text-muted"><?= $drink['price']?> Đ</div>
                </div>
            </div>
        </div>
        <?php
    }
}
>>
```

Hình 2.96: Mã nguồn hiển thị trang đồ uống

Ảnh sản phẩm được bọc bởi thẻ **<a>** và có đường dẫn (**href**) gọi tới controller **'drink'** và hành động **'detail'**, kèm theo đó là biến **'id'** có giá trị bằng với giá trị ID của đồ uống tương ứng. Controller sẽ lấy đồ uống theo id bằng cách dùng hàm **getDataByID(\$tblTable,\$id)**, chúng ta cần truyền vào 2 giá trị đó là tên bảng và **'id'** của đồ uống cần hiển thị trên trang chi tiết sản phẩm.

```
case 'detail':{
    if (isset($_GET['id'])) {
        $id = $_GET['id'];
        $drinkByID = $db->getDataByID($tblTable,$id);
    }
    require_once('view/drink/detail_drink.php');
    break;
}
```

Hình 2.97: Mã nguồn chức năng hiển thị đồ uống theo thẻ loại

Hàm **getDataByID()** có chức năng thực thi câu lệnh sql ***SELECT * FROM 'drink' WHERE ID = \$id*** để lấy một đồ uống với ID cụ thể.

```
//lay 1 row
public function getDataByID($table, $id)
{
    $sql = "SELECT * FROM $table where ID = $id";
    $this->execute($sql);
    if ($this->countRow() != 0) {
        $data = mysqli_fetch_array($this->result);
    }else{
        $data = 0;
    }
    return $data;
}
```

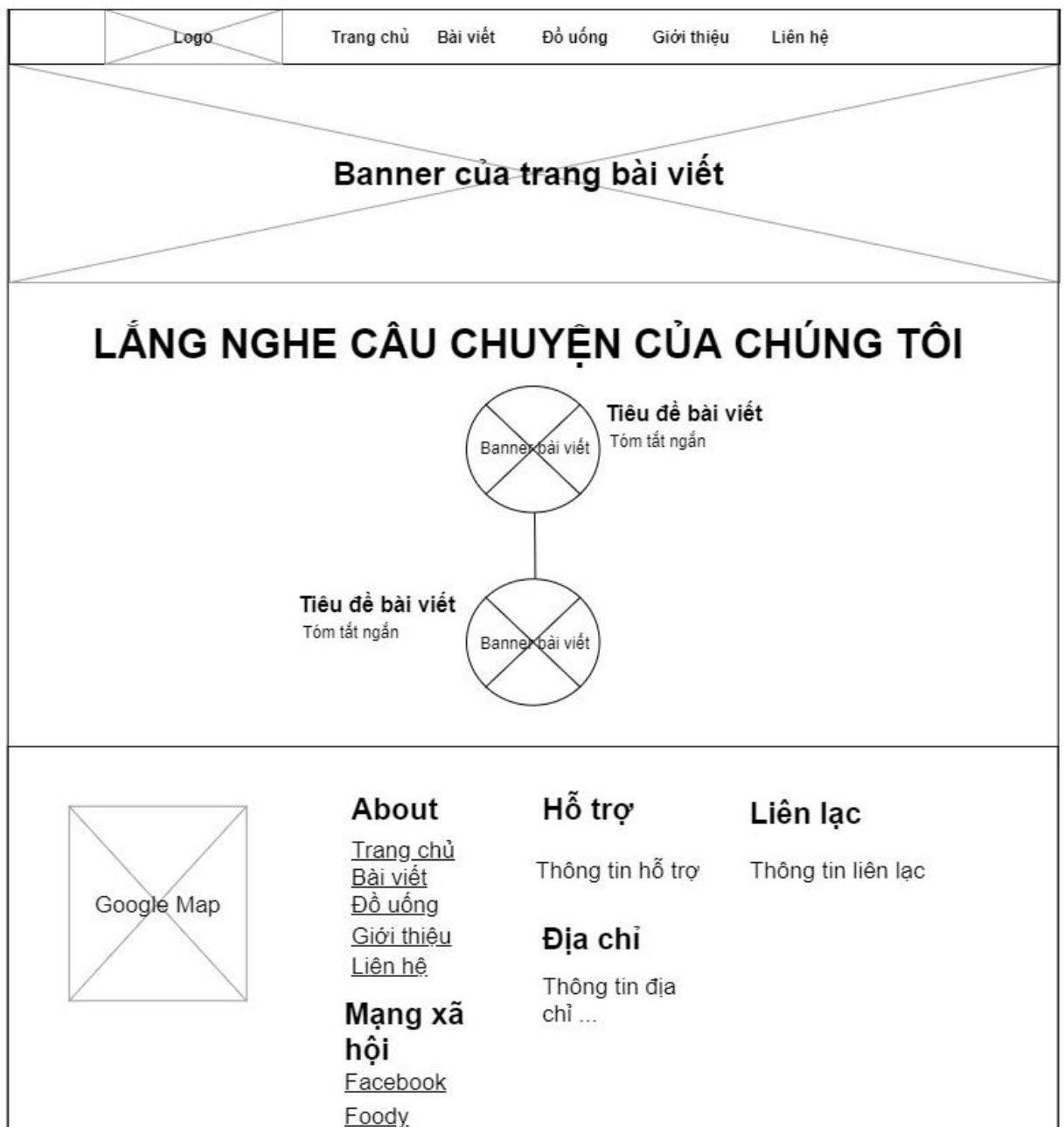
Hình 2.98: Mã nguồn để lấy dữ liệu đồ uống từ trong cơ sở dữ liệu

Ảnh dưới đây là giao diện được thiết kế của trang chi tiết đồ uống, hiển thị chi tiết các thông tin của đồ uống như tên, giá và mô tả.

Logo		Trang chủ	Bài viết	Đồ uống	Giới thiệu	Liên hệ
 Hình ảnh đồ uống		Tên đồ uống Giá đồ uống <u>MUA NGAY</u> (link tới trang Foody) Gọi giao hàng tận nơi: 0914619188				
 Google Map	About		Hỗ trợ		Liên lạc	
	Trang chủ		Thông tin hỗ trợ		Thông tin liên lạc	
	Bài viết					
	Đồ uống					
	Giới thiệu					
Liên hệ		Địa chỉ				
		Thông tin địa chỉ ...				
Mạng xã hội						
Facebook						
Foody						

Hình 2.99: Thiết kế giao diện trang chi tiết đồ uống

4.4. Trang danh sách bài viết



Hình 2.100: Thiết kế giao diện trang danh sách bài viết

Khi người dùng click vào ‘ Bài viết’ controller post sẽ được gọi:

```
index.php
<?php
include "model/drink.php";
$db = new Drink;
$db->connect();
$tblTable = 'post';
$subTable1 = 'image';
$posts = $db->getAllData($tblTable);
$images = $db->getAllData($subTable1);
if (isset($_GET['action'])) {
    $action = $_GET['action'];
}else{
    $action = '';
}

switch ($action) {
    case 'list':{
        require_once('view/post/list_post.php');
        break;
    }
    case 'detail':{
        if (isset($_GET['id'])) {
            $id = $_GET['id'];
            $postByID = $db->getDataByID($tblTable,$id);
        }
        require_once('view/post/detail_post.php');
        break;
    }
}
```

Hình 2.101: Mã nguồn giao diện trang danh sách bài viết

Khi đã lấy được dữ liệu trong các bảng bằng cách dùng hàm **getAllData()**, ta sẽ hiển thị chúng ra ngoài màn hình bằng cách dùng vòng lặp **foreach()** :

```
<?php
foreach ($posts as $post){
    <li>
        <div class="timeline-image" style="height: 100%;">
            <div id="postThumbnail" class="owl-theme" style="height: 100%;">
                <?php
                foreach ($images as $image) {
                    if ($image['bannerID'] == $post['bannerID']) {
                        <div class="item" style="padding:unset;height: 100%;">
                            
                        </div>
                    }
                }
                <?php
            </div>
        </div>
        <div class="timeline-panel">
            <div class="timeline-heading">
                <?=$post['dateCreate']>
                <a href="http://localhost.com/helentea/index.php?controller=post&action=detail&id=<?=$post['ID']">>
                <h4 class="subheading"><?=$post['name']></h4>
            </div>
            <div class="timeline-body"><p class="text-muted"><?=$post['shortDescription']></p></div>
        </div>
    </li>
}
<?php
```

Hình 2.102: Mã nguồn chức hiển thị danh sách bài viết

Ảnh bài viết được bọc bởi thẻ `<a></a>` và có đường dẫn (**href**) gọi tới controller ‘post’ và hành động ‘detail’, kèm theo đó là biến ‘id’ có giá trị bằng với giá trị ID của bài viết tương ứng. Controller sẽ hiển thị bài viết theo id bằng cách dùng hàm **getDataByID(\$tblTable,\$id)**, chúng ta cần truyền vào 2 giá trị đó là tên bảng và ‘id’ của bài viết cần hiển thị trên trang chi tiết bài viết.

```
case 'detail':{
    if (isset($_GET['id'])) {
        $id = $_GET['id'];
        $postById = $db->getDataByID($tblTable,$id);
    }
    require_once('view/post/detail_post.php');
    break;
}
```

Hình 2.103: Mã nguồn chức năng hiển thị bài viết

Hàm **getDataByID()** có chức năng thực thi câu lệnh sql ***SELECT * FROM 'post' WHERE ID = \$id*** để lấy một bài viết với ID cụ thể.

```
//lay 1 row
public function getDataByID($table, $id)
{
    $sql = "SELECT * FROM $table where ID = $id";
    $this->execute($sql);
    if ($this->countRow() != 0) {
        $data = mysqli_fetch_array($this->result);
    }else{
        $data = 0;
    }
    return $data;
}
```

Hình 2.104: Mã nguồn chức năng lấy bài viết từ trong cơ sở dữ liệu

Ảnh dưới đây là giao diện của trang chi tiết bài viết, hiển thị chi tiết các thông tin của bài viết như tên bài viết, mô tả ngắn, nội dung và Banner Slider được gắn đầu bài viết.



Hình 2.105: Thiết kế giao diện trang chi tiết bài viết

4.5. Banner Slider

Do website đang sử dụng jQuery nên để rút ngắn thời gian cũng như tài nguyên, em sẽ sử dụng Owl Carousel 2, là một plugin được xây dựng trên thư viện jQuery hỗ trợ chúng ta tạo ra những Banner Slider đẹp và hỗ trợ luôn cả responsive.

Để sử dụng plugin này, ta cần sử dụng một số file sau và import chúng vào file index.php.

```
/* jquery-1.11.1.min.js
/* owl.carousel.js
/* owl.carousel.min.js
```

Hình 2.106: Danh sách file jQuery được import vào file index.php

```
<script src="pub/js/owl.carousel.min.js"></script>
<link rel="stylesheet" href="pub/css/owl.carousel.min.css">
<link rel="stylesheet" href="pub/css/owl.theme.default.css">
```

Hình 2.107: Mã nguồn link file jQuery được import vào file index.php

Sau bước này, chúng ta có thể tạo ra một slider rất đơn giản bằng cách sử dụng đoạn mã html sau:

```
<!-- Set up your HTML -->
<div class="owl-carousel">
  <div> Your Content </div>
  <div> Your Content </div>
  <div> Your Content </div>
  <div> Your Content </div>
  <div> Your Content </div>
  <div> Your Content </div>
  <div> Your Content </div>
  <div> Your Content </div>
</div>
```

Hình 2.108: Mã nguồn để tạo BannerSlider

Sau đó gọi ra chức năng của plugin Owl Carousel

```
$(document).ready(function(){
  $(".owl-carousel").owlCarousel();
});
```

Hình 2.109: Mã nguồn chức năng plugin Owl Carousel

4.6. Footer



Hình 2.110: Thiết kế giao diện footer

- Dưới đây là mã nguồn của giao diện footer:

```
<div class="col-3 footer-element" >
  <h2 class="footer_title">ABOUT</h2>
  <ul class="footer_linklists">
    <li><a href="http://localhost/helentea/index.php" title="Trang chủ">Trang chủ</a>
    </li>
    <li><a href="http://localhost/helentea/index.php?controller=post&action=list"
    title="Bài viết">Bài viết</a></li>
    <li><a href="http://localhost/helentea/index.php?controller=drink&action=list"
    title="Đồ uống">Đồ uống</a></li>
    <li><a href="http://localhost/helentea/index.php?controller=home&action=about"
    title="Giới thiệu">Giới thiệu</a></li>
    <li><a href="http://localhost/helentea/index.php?controller=home&action=contact"
    title="Liên hệ">Liên hệ</a></li>
  </ul>
  <div class="address_footer" style="padding-top: 10px">
    <h2 class="footer_title">Social</h2>
    <a style="float: right;" href="https://www.foody.vn/hai-phong/hellen-tea">
      
    </a>
    <a style="float: right; padding-right: 15px" href="https://www.facebook.com/Hellen-Tea-116996860034160/">
      
    </a>
  </div>
</div>
```

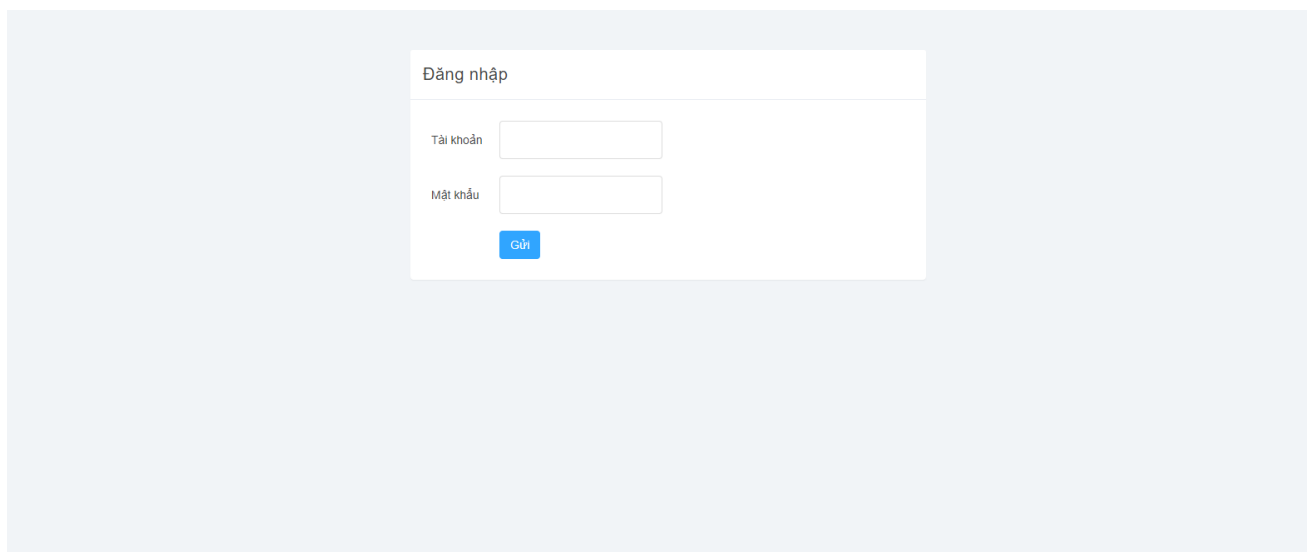
Hình 2.111: Mã nguồn giao diện footer

Em dùng **href** để gán những trang như: Trang chủ, Bài viết, Đồ uống, Giới thiệu, Liên hệ để người dùng có thể chuyển đến những trang này dù đang xem Footer của website.

Em dùng **href** để gán trang mạng xã hội Facebook và Foody của quán. Khi bấm vào logo của từng phần, nó sẽ chuyển người dùng tới đó.

CHƯƠNG III: DEMO WEBSITE

1. Demo Back-end

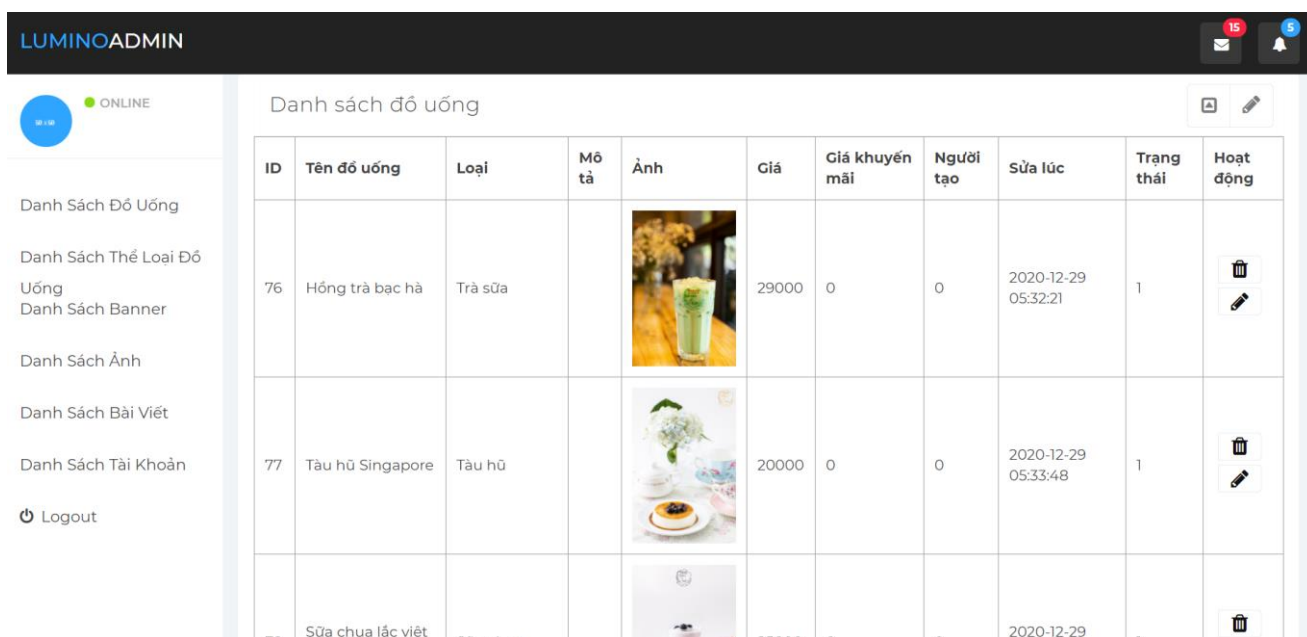


Đăng nhập

Tài khoản

Mật khẩu

Hình 3.1: Giao diện đăng nhập



LUMINOADMIN

ONLINE

Danh Sách Đồ Uống

Danh Sách Thể Loại Đồ Uống

Danh Sách Banner

Danh Sách Ảnh

Danh Sách Bài Viết

Danh Sách Tài Khoản

Logout

Danh sách đồ uống

ID	Tên đồ uống	Loại	Mô tả	Ảnh	Giá	Giá khuyến mãi	Người tạo	Sửa lúc	Trạng thái	Hoạt động
76	Hồng trà bạc hà	Trà sữa			29000	0	0	2020-12-29 05:32:21	1	 
77	Tàu hũ Singapore	Tàu hũ			20000	0	0	2020-12-29 05:33:48	1	 
78	Sữa chua lắc việt	Sữa chua			25000	0	0	2020-12-29	1	 

Hình 3.2: Giao diện trang Admin (Back-end)

Giao diện đăng nhập với vai trò Admin

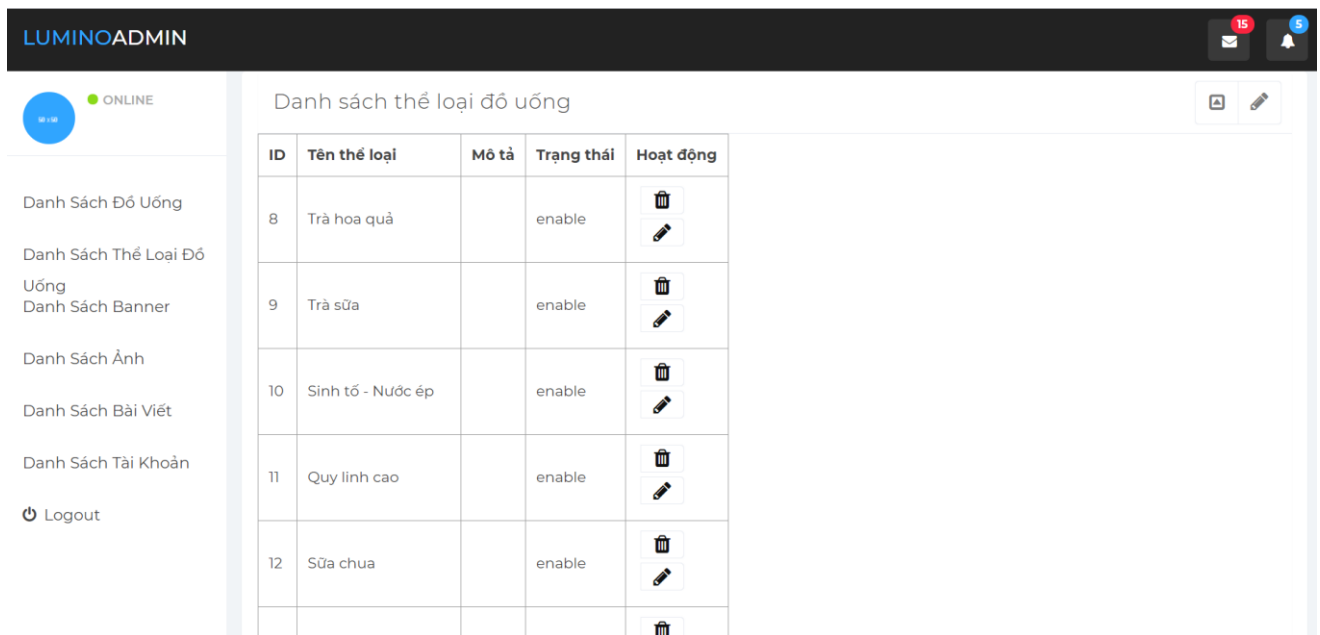
Danh sách các menu gồm danh mục:

- Danh sách đồ uống
- Danh sách thể loại đồ uống
- Danh sách banner
- Danh sách ảnh
- Danh sách bài viết
- Danh sách tài khoản
- Logout
-

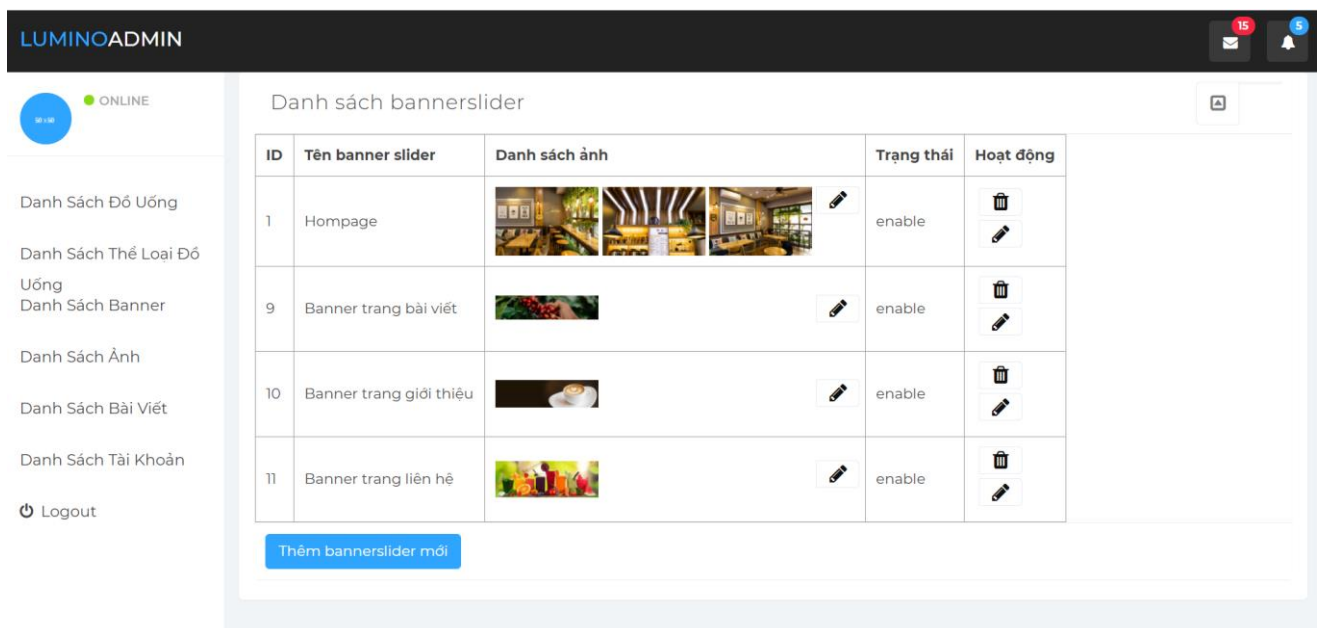
The screenshot shows the LUMINOADMIN interface. On the left is a sidebar with a user profile (ONLINE) and navigation links: Danh Sách Đồ Uống, Danh Sách Thể Loại Đồ Uống, Danh Sách Banner, Danh Sách Ảnh, Danh Sách Bài Viết, Danh Sách Tài Khoản, and Logout. The main content area is titled 'Danh sách đồ uống' and contains a table with the following data:

ID	Tên đồ uống	Loại	Mô tả	Ảnh	Giá	Giá khuyến mãi	Người tạo	Sửa lúc	Trạng thái	Hoạt động
76	Hồng trà bạc hà	Trà sữa			29000	0	0	2020-12-29 05:32:21	1	
77	Tàu hũ Singapore	Tàu hũ			20000	0	0	2020-12-29 05:33:48	1	
78	Sữa chua lác việt	Sữa chua			25000	0	0	2020-12-29	1	

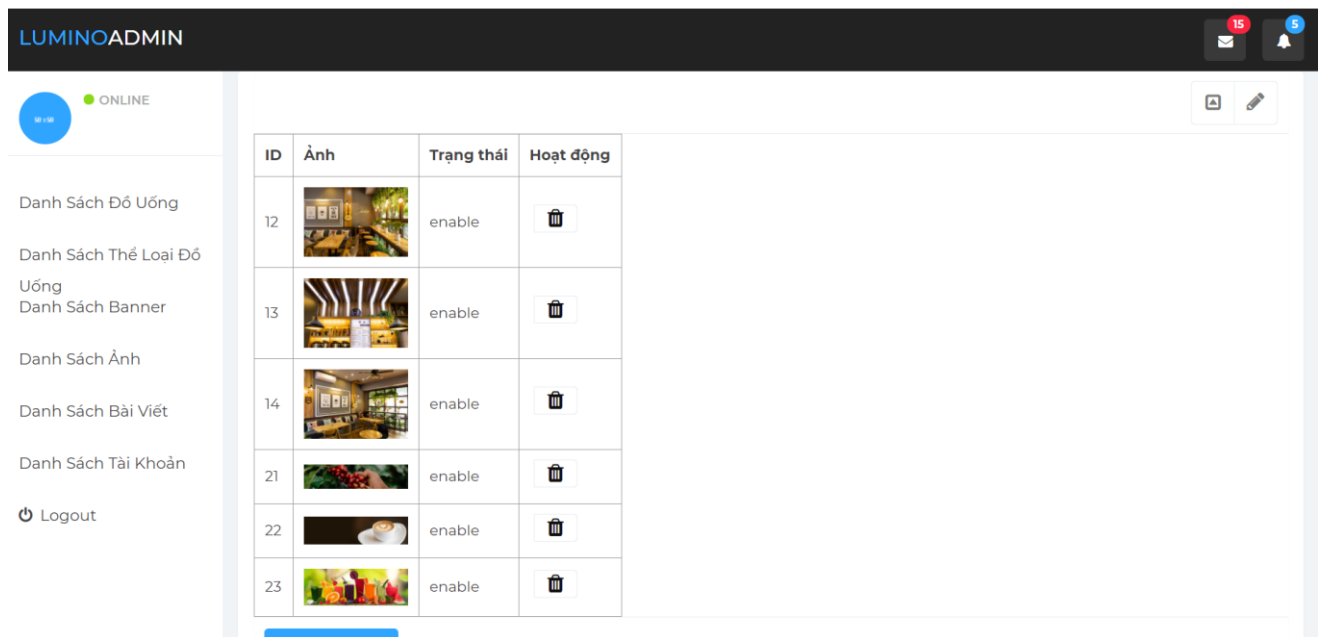
Hình 3.3: Giao diện trang danh sách đồ uống



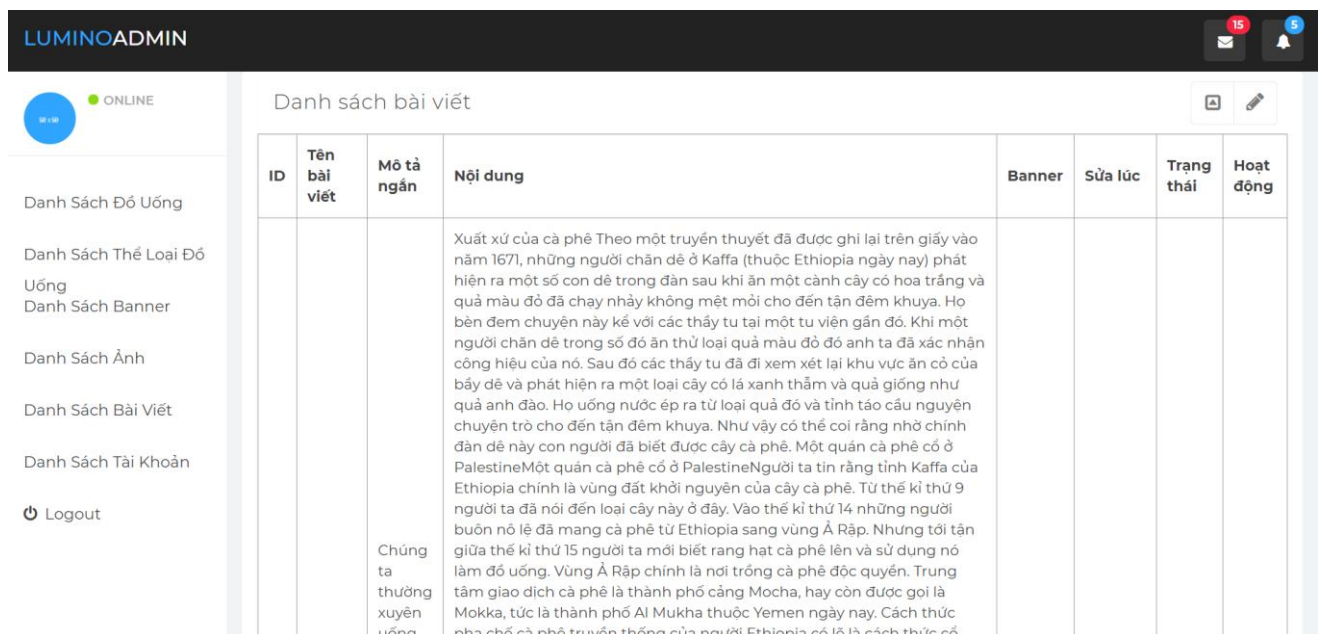
Hình 3.4: Giao diện trang danh sách thể loại đồ uống



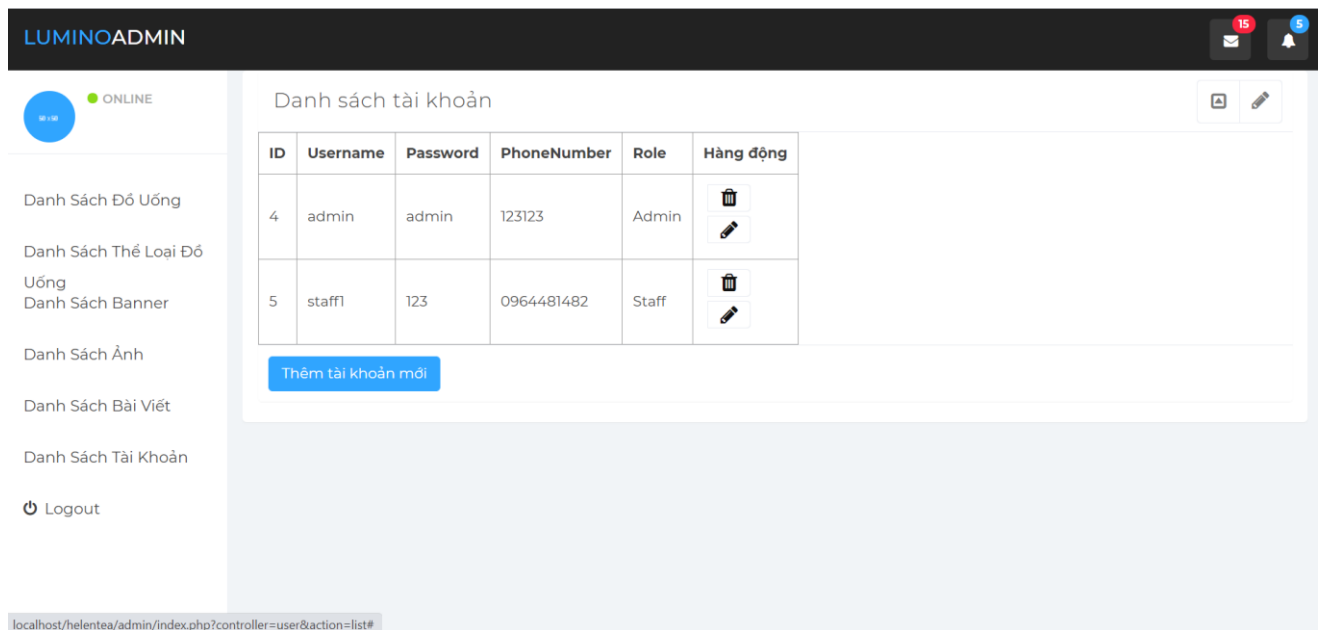
Hình 3.5: Giao diện trang danh sách banner



Hình 3.6: Giao diện trang danh sách ảnh

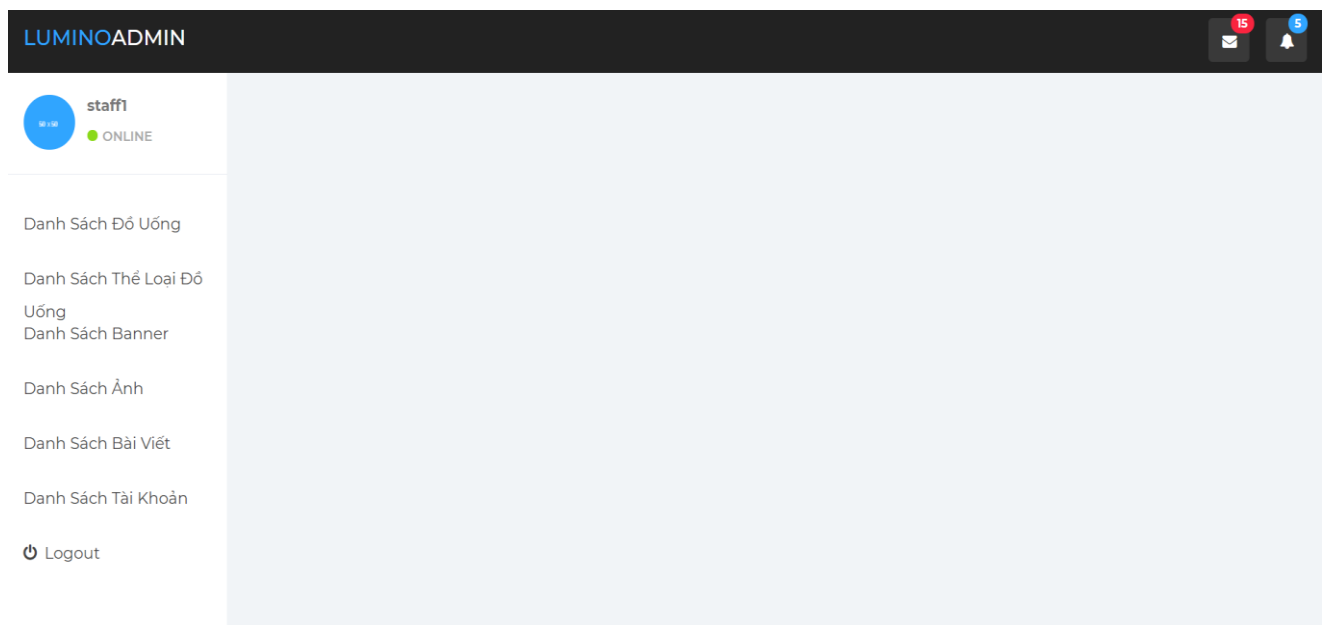


Hình 3.7: Giao diện trang danh sách bài viết



Hình 3.8: Giao diện trang danh sách tài khoản

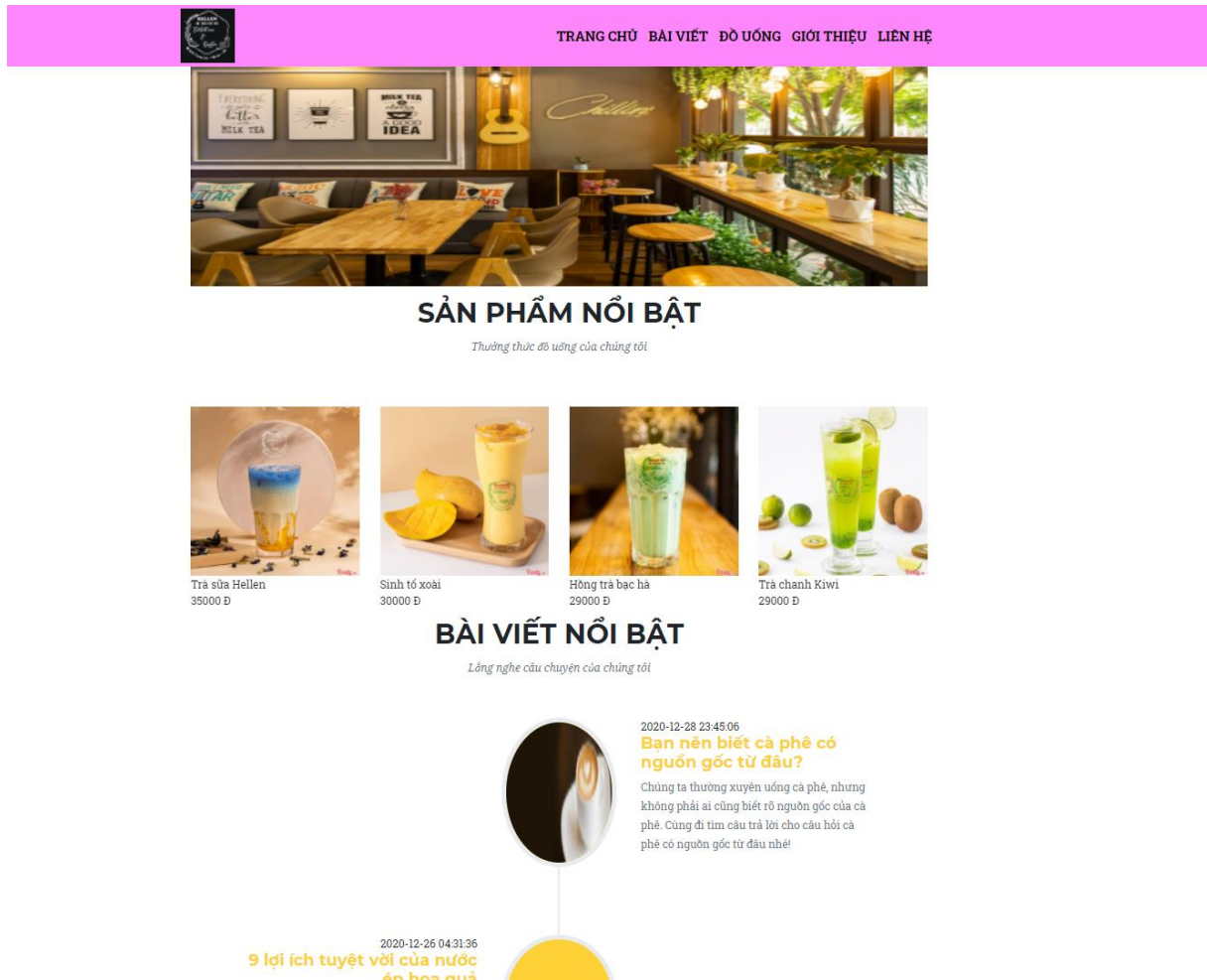
Giao diện khi người dùng đăng nhập với vai trò Staff (Nhân viên) thì khi click vào mục “Danh sách tài khoản”, nhân viên sẽ không thể xem cũng như thêm sửa xóa được tài khoản.



Hình 3.9: Giao diện trang danh sách tài khoản khi đăng nhập với vai trò Staff

2. Demo Front-end

2.1. Demo trang chủ



Hình 3.10: Giao diện trang chủ

- Tại giao diện trang chủ gồm logo quán, thanh điều hướng với danh sách Menu gồm các trang:

- Trang chủ
- Bài viết
- Đồ uống
- Giới thiệu
- Liên hệ

- Giao diện trang chủ có banner, danh sách sản phẩm nổi bật, bài viết nổi bật.



Hình 3.11: Giao diện footer

- Giao diện footer hiển thị bản đồ chỉ dẫn tới quán và các thông tin liên quan.
- Tại mục social hiển thị logo link tới trang Facebook, Foody của Hellen Tea & Coffee.

2.2. Demo trang bài viết



Hình 3.12: Giao diện trang bài viết



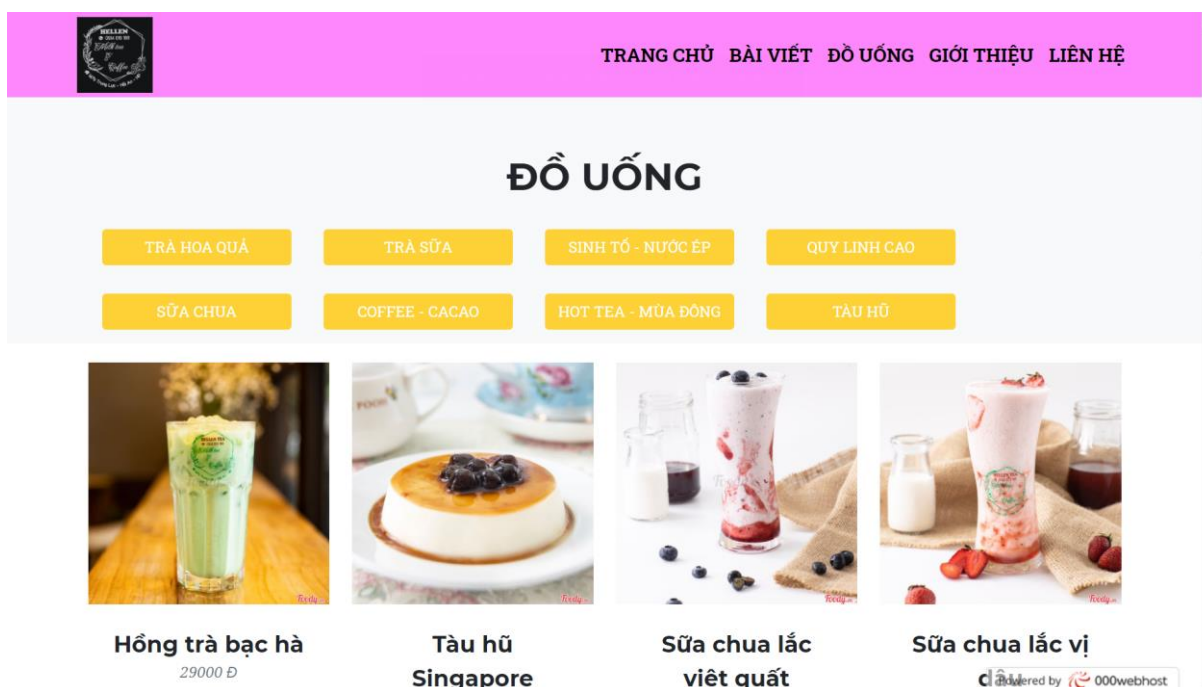
Bạn nên biết cà phê có nguồn gốc từ đâu?

Chúng ta thường xuyên uống cà phê, nhưng không phải ai cũng biết rõ nguồn gốc của cà phê. Cùng đi tìm câu trả lời cho câu hỏi cà phê có nguồn gốc từ đâu nhé!

Xuất xứ của cà phê Theo một truyền thuyết đã được ghi lại trên giấy vào năm 1671, những người chăn dê ở Kaffa (thuộc Ethiopia ngày nay) phát hiện ra một số con dê trong đàn sau khi ăn một cành cây có hoa trắng và quả màu đỏ đã chạy nhảy không mệt mỏi cho đến tận đêm khuya. Họ bèn đem chuyện này kể với các thầy tu tại một tu viện gần đó. Khi một người chăn dê trong số đó ăn thử loại quả màu đỏ đó anh ta đã xác nhận công hiệu của nó. Sau đó các thầy tu đã đi xem xét lại khu vực ăn cỏ của bầy dê và phát hiện ra một loại cây có lá xanh thẫm và quả giống như quả anh đào. Họ uống nước ép ra từ loại quả đó và tinh tảo cầu nguyện chuyện trò cho đến tận đêm khuya. Như vậy có thể coi rằng nhờ chính

Hình 3.13: Giao diện trang chi tiết bài viết

2.3. Demo trang đồ uống



Hình 3.14: Giao diện trang đồ uống



Trà sữa Hellen

35000 đ

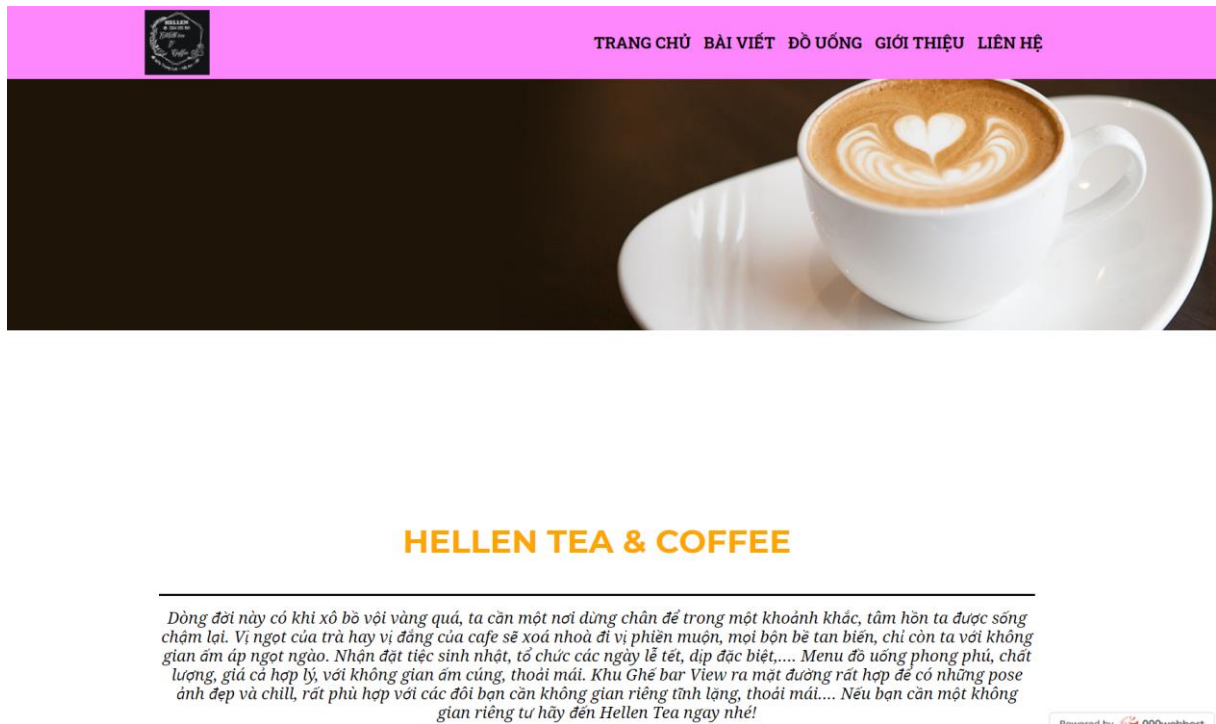
MUA NGAY

Gọi giao hàng tận nơi **0914619188**

Hình 3.15: Giao diện trang chi tiết đồ uống

- Nhấn vào **“MUA NGAY”** sẽ dẫn tới trang Foody của quán để giúp khách hàng đặt mua sản phẩm
- Nhấn **“Gọi giao hàng tận nơi”** để gọi trực tiếp đến quán.

2.4. Demo trang giới thiệu



Hình 3.16: Giao diện trang giới thiệu

2.5. Demo trang liên hệ



Hình 3.17: Giao diện trang liên hệ

KẾT LUẬN

Về cơ bản em đã hiểu được quá trình xây dựng một Website bằng một ngôn ngữ lập trình mới. Xây dựng được trang back-end với giao diện thân thiện, từ ngữ dễ hiểu, dễ sử dụng với người dùng. Đồng thời tạo được các chức năng quản lý nội dung khiến người dùng dễ dàng thao tác, và quản lý. Xây dựng chức năng phân quyền người dùng khi đăng nhập để quản lý nội dung. Xây dựng được trang front-end giúp cung cấp các thông tin cần thiết của một website về quán cafe. Xây dựng website hiển thị nội dung linh hoạt, giúp quán cập nhật sản phẩm, hình ảnh, bài viết dễ dàng. Thiết kế giao diện với màu sắc và bố cục hài hòa tạo cho người sử dụng cảm thấy thoải mái khi sử dụng.

Tuy nhiên do thời gian có hạn nên kết quả thực hiện vẫn còn một số hạn chế, website còn nhỏ, chưa có chức năng tìm kiếm nội dung, chức năng đăng ký, đăng nhập, và chức năng chat trực tuyến với người dùng. Trong thời gian sắp tới em sẽ tiếp tục phát triển website để đáp ứng được các yêu cầu cấp thiết và sử dụng thuận tiện cho người dùng.

TÀI LIỆU THAM KHẢO

- [1]. Phạm Hữu Khang – Lập trình web bằng php 5.3 và MySQL 5.1 – NXB Phương Đông.
- [2]. Nguyễn Văn Vị (2002), Phân tích và thiết kế hệ thống thông tin quản lý, NXB Thống kê.
- [3]. Lê Văn Phùng (2010), CSDL quan hệ và công nghệ phân tích – thiết kế, NXB Thông tin và Truyền thông.
- [4]. <https://www.w3schools.com/>
- [5]. <https://getbootstrap.com/docs/5.0/layout/columns/>