**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC SAO ĐỎ**

****

**BÁO CÁO**

**ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP CƠ SỞ   
NĂM HỌC 2019 - 2020**

**TÊN ĐỀ TÀI:**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ SẢN XUẤT**

**NƯỚC UỐNG TINH KHIẾT ĐÓNG BÌNH TẠI TRƯỜNG ĐẠI HỌC SAO ĐỎ**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **: Lê Đức Minh** (chủ nhiệm)  **: Nguyễn Quang Nguyên** |
| **Giảng viên hướng dẫn** | **: Phạm Văn Kiên** |
| **Đơn vị** | **: Khoa Điện tử - Tin học** |

**LỜI NÓI ĐẦU**

Trong những năm gần đây, công nghệ thông tin hiện nay đã có những bước phát triển mạnh mẽ và đạt được nhiều thành tựu.Máy tính điện tử đã không còn là vật quý hiếm nữa mà được sử dụng rộng rãi trên phạm vi toàn cầu, nó nhanh chóng ăn sâu và giữ vai trò chủ đạo trong các ngành kinh tế, quân sự, giáo dục,…và ngày càng trở nên thông dụng, gần gũi với con người. Đứng trước sự bùng nổ thông tin, các cá nhân, các tổ chức và doanh nghiệp đều tìm biện pháp để hoàn thiện hệ thống thông tin của mình nhằm nâng cao hiệu quả công việc và giảm được các chi phí. Do đó việc tin học hóa các hoạt động của các hệ thống đã cũ và không hiệu quả là việc làm hết sức cần thiết.

Việc ứng dụng Tin học vào các lĩnh vực quản lí hết sức cần thiết trong quá trình làm việc để đạt được hiệu quả cao cũng như những lợi ích trong quá trình quản lí. Sự lớn mạnh của máy vi tính đã giúp đỡ con người rất nhiều trong công việc đặc biệt là trong công tác quản lý, nghiên cứa khoa học…Thông qua máy vi tính con người có thể xử lý những công việc phức tạp mà con người không thể thực hiện thủ công được hoặc rất khó khăn trong quá trình thực hiện.

Từ nhu cầu cần thiết phải tin học hóa các hệ thống thông tin như trên, em sử dụng các kiến thức đã học trong nhà trường tìm hiểu và đi sâu vào phân tích quy trình Quản lý sản xuất nước tại cơ sở trường Đại học Sao Đỏ để: **“Xây dựng phần mềm quản lý sản xuất nước uống tình khiết đóng bình tại trường Đại học Sao Đỏ”**.

Dựa trên ngôn ngữ lập trình PHP cùng với hệ quản trị cơ sở MySQL ứng dụng đã giải quyết được những vấn đề của bài toán, giúp nâng cao hiệu quả quản lý sản xuất cho xưởng sản xuất nước.

# MỤC LỤC

[MỤC LỤC 1](#_Toc43566111)

[DANH MỤC CÁC BẢNG 3](#_Toc43566112)

[DANH MỤC CÁC HÌNH 4](#_Toc43566113)

[MỞ ĐẦU 5](#_Toc43566114)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 7](#_Toc43566115)

[1.1 Giới thiệu về ngôn ngữ PHP 7](#_Toc43566116)

[1.1.1 Giới thiệu chung về ngôn ngữ PHP 7](#_Toc43566117)

[1.1.2 Các tính năng của PHP 7](#_Toc43566118)

[1.1.3 Phiên bản PHP 7 8](#_Toc43566119)

[1.1.4 Làm việc với PHP bằng xampp 8](#_Toc43566120)

[1.2 Giới thiệu về cơ sở dữ liệu MySQL 9](#_Toc43566121)

[1.2.1 Tổng quan về MySQL 9](#_Toc43566122)

[1.2.2 Các phiên bản MySQL 10](#_Toc43566123)

[1.2.3 Kiểu dữ liệu và thuộc tính 10](#_Toc43566124)

[1.2.4 Sử dụng MySQL bằng xampp 12](#_Toc43566125)

[1.3 Giới thiệu về kiến trúc MVC 13](#_Toc43566126)

[1.3.1 Tổng quan về kiến trúc MVC 13](#_Toc43566127)

[1.3.2 Ưu điểm của kiến trúc MVC 14](#_Toc43566128)

[1.3.3 Framewok Codeigniter 15](#_Toc43566129)

[1.3.4 Bootstrap 16](#_Toc43566130)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 18](#_Toc43566131)

[2.1 Khảo sát hệ thống 18](#_Toc43566132)

[2.1.1 Hệ thống cũ 18](#_Toc43566133)

[2.1.2 Đề xuất xây dựng hệ thống mới 19](#_Toc43566134)

[2.2 Phân tích xây dựng hệ thống mới 19](#_Toc43566135)

[2.2.1 Xác định các tác nhân 19](#_Toc43566136)

[2.2.2 Đặc tả yêu cầu 19](#_Toc43566137)

[2.3 Xây dựng các biểu đồ 21](#_Toc43566138)

[2.3.1 Biểu đồ phân cấp chức năng 21](#_Toc43566139)

[2.3.2 Biểu đồ luồng dữ liệu 22](#_Toc43566140)

[2.4 Phân tích thiết kế cơ sở dữ liệu 22](#_Toc43566141)

[2.4.1 Xác định các thực thể 22](#_Toc43566142)

[2.4.2 Mô tả chi tiết các thực thể 23](#_Toc43566143)

[2.4.3 Sơ đồ liên kết các thực thể 27](#_Toc43566144)

[2.5 Áp dụng mô hình MVC và ưu điểm 27](#_Toc43566145)

[2.5.1 Mô hình MVC trong xây dựng chương trình 27](#_Toc43566146)

[2.5.2 Ưu điểm khi sử dụng mô hình MVC 31](#_Toc43566147)

[CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH 33](#_Toc43566148)

[3.1 Thiết kế cơ sở dữ liệu với MySQL 33](#_Toc43566149)

[3.1.1 Bảng quản lý tài khản (user) 33](#_Toc43566150)

[3.1.2 Bảng loại sản phẩm (loaisanpham) 33](#_Toc43566151)

[3.1.3 Bảng sản phầm (sanpham) 33](#_Toc43566152)

[3.1.4 Bảng hoá đơn (hoadon) 33](#_Toc43566153)

[3.1.5 Bảng chi tiết hoá đơn (chitiethoadon) 33](#_Toc43566154)

[3.1.6 Bảng nhập hàng (nhaphang) 34](#_Toc43566155)

[3.1.7 Bảng chi tiết nhập hàng (chitietnhaphang) 34](#_Toc43566156)

[3.1.8 Bảng sản xuất (sanxuat) 34](#_Toc43566157)

[3.1.9 Bảng chi tiết sản xuất (chitietsanxuat) 34](#_Toc43566158)

[3.1.10 Bảng khách hàng 34](#_Toc43566159)

[3.2 Thiết kế chức năng 35](#_Toc43566160)

[3.2.1 Trang chủ 35](#_Toc43566161)

[3.2.2 Chức năng đăng nhập 37](#_Toc43566162)

[3.2.3 Chức năng quản lý tài khoản 39](#_Toc43566163)

[3.2.4 Chức năng quản lý loại sản phẩm 43](#_Toc43566164)

[3.2.5 Chức năng quản lý sản phẩm 45](#_Toc43566165)

[3.2.6 Chức năng quản lý hoá đơn 48](#_Toc43566166)

[3.2.7 Chức năng quản lý nhập hàng 53](#_Toc43566167)

[3.2.8 Chức năng quản lý sản xuất 56](#_Toc43566168)

[3.2.9 Chức năng thống kê 59](#_Toc43566169)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 61](#_Toc43566170)

# DANH MỤC CÁC BẢNG

[Bảng 2.1 Thực thể USER 23](#_Toc43566070)

[Bảng 2.2 Thực thể KHACHHANG 24](#_Toc43566071)

[Bảng 2.3 Thực thể SANPHAM 24](#_Toc43566072)

[Bảng 2.4 Thực thể LOAISANPHAM 24](#_Toc43566073)

[Bảng 2.5 Thực thể HOADON 25](#_Toc43566074)

[Bảng 2.6 Thực thể CHITIETHOADON 25](#_Toc43566075)

[Bảng 2.7 Thực thể NHAPHANG 25](#_Toc43566076)

[Bảng 2.8 Thực thể CHITIETNHAPHANG 26](#_Toc43566077)

[Bảng 2.9 Thực thể SANXUAT 26](#_Toc43566078)

[Bảng 2.10 Thực thể CHITIETSANXUAT 26](#_Toc43566079)

# DANH MỤC CÁC HÌNH

[Hình 2.1 Biểu đồ phân cấp chức năng 21](#_Toc43566080)

[Hình 2.2 Biểu đồ luồng dữ liệu mức khung cảnh 22](#_Toc43566081)

[Hình 2.3 Biểu đồ luồng dữ liệu mức đỉnh 22](#_Toc43566082)

[Hình 2.4 Sơ đồ liên kết các thực thể 27](#_Toc43566083)

[Hình 3.1 Bảng user 33](#_Toc43566084)

[Hình 3.2 Bảng loaisanpham 33](#_Toc43566085)

[Hình 3.3 Bảng sanpham 33](#_Toc43566086)

[Hình 3.4 Bảng hoadon 33](#_Toc43566087)

[Hình 3.5 Bảng chitiethoadon 33](#_Toc43566088)

[Hình 3.6 Bảng nhaphang 34](#_Toc43566089)

[Hình 3.7 Bảng chitietnhaphang 34](#_Toc43566090)

[Hình 3.8 Bảng sanxuat 34](#_Toc43566091)

[Hình 3.9 Bảng chitietsanxuat 34](#_Toc43566092)

[Hình 3.10 Bảng khachhang 34](#_Toc43566093)

[Hình 3.11 Giao diện trang chủ trên máy tính 35](#_Toc43566094)

[Hình 3.12 Giao diện trang chủ trên màn hình điện thoại 35](#_Toc43566095)

[Hình 3.13 Trang đăng nhập 37](#_Toc43566096)

[Hình 3.14 Trang thông tin tài khoản 39](#_Toc43566097)

[Hình 3.15 Trang thêm tài khoản 40](#_Toc43566098)

[Hình 3.16 Trang sửa thông tin tài khoản 40](#_Toc43566099)

[Hình 3.17 Trang quản lý loại sản phẩm 43](#_Toc43566100)

[Hình 3.18 Trang quản lý sản phẩm 45](#_Toc43566101)

[Hình 3.19 Trang thêm sản phẩm mới 46](#_Toc43566102)

[Hình 3.20 Trang quản lý hoá đơn 48](#_Toc43566103)

[Hình 3.21 Trang thêm hoá đơn mới 48](#_Toc43566104)

[Hình 3.22 Trang chỉnh sửa hoá đơn 49](#_Toc43566105)

[Hình 3.23 Trang quản lý nhập hàng 53](#_Toc43566106)

[Hình 3.24 Trang thêm đơn nhập hàng 53](#_Toc43566107)

[Hình 3.25 Trang thêm đơn sản xuất 56](#_Toc43566108)

[Hình 3.26 Trang thống kê 59](#_Toc43566109)

[Hình 3.27 Ví dụ thống kê hoá đơn 59](#_Toc43566110)

# MỞ ĐẦU

1. **Lí do nghiên cứu đề tài**

Ngày nay cùng với sự phát triển như vũ bão của nền kinh tế thị trường và sự gia tăng mạnh về nhu cầu thiết yếu của con người thì những cửa hàng buôn bán cũng xuất hiện và phát triển ngày càng nhiều với quy mô đa dạng. Sự phát triển ấy đòi hỏi một đội ngủ quản lý thật chuyên nghiệp, có trình độ chuyên môn sâu và phải kết hợp phương pháp quản lý hiệu quả.

Trước đây, việc thu thập thông tin và quản lý hàng hóa được thực hiện bằng những phương pháp thủ công thông qua ghi chép sổ sách, quá trình tính toán số liệu nhập xuất thường hay nhầm lẫn sai sót và tốn nhiều thời gian. Do đó với khối lượng công việc lớn như vậy thì các phương pháp quản lý truyền thống sẽ tốn nhiều thời gian, công sức và khó có thể đáp ứng được các nhu cầu ngày càng cao của con người. Hiện nay công nghệ thông tin ngày càng phát triển và ngày càng được ứng dụng vào thực tế, việc áp dụng công nghệ vào trong quá trình quản lý, góp phần đơn giản hóa việc quản lý, thao tác với dữ liệu. Vì vậy việc kết hợp và áp dụng các phương pháp quản lý hiện đại vào việc quản lý bán hàng giúp con người quản lý thông tin một cách chính xác, dễ dàng, nhanh chóng, khoa học, giảm bớt hao tổn công sức, chi phí và tăng cường hoạt động bán hàng.

Phần mềm quản lý sản xuất nước uống tinh khiết đóng bình nhằm giúp đơn giản và tối ưu hóa các công đoạn quản lý sản xuất các bình nước uống tinh khiết từ quản lý khách hàng, đơn hàng đến lập kế hoạch giao hàng một cách tự động.

1. **Mục tiêu nghiên cứu**

Nghiên cứu, xây dựng phần mềm quản lý sản xuất nước uống tinh khiết đóng bình, áp dụng cho công tác quản lý sản xuất tại trường Đại học Sao Đỏ.

1. **Đối tượng nghiên cứu**

Các tài liệu, quy định, quy trình về quản lý sản xuất nước uống tinh khiết đóng bình tại trường Đại học Sao Đỏ.

1. **Nội dung nghiên cứu**

- Quản lý khách hàng;

- Quản lý đơn đặt hàng, tự động cập nhật đơn đặt hàng cố định của khách đặt dài hạn;

- Quản lý giao hàng, tự động cập nhật lịch giao hàng từ đơn đặt hàng;

- Quản lý vỏ bình, cập nhật vỏ bình tồn;

- Quản lý công nợ, tự động cập nhật số phải thu;

- Quản lý vật tư đầu vào.

1. **Phương pháp nghiên cứu**

* Phương pháp nghiên cứu: Nghiên cứu tài liệu, thu thập dữ liệu thực tế, lập trình thực nghiệm.
* Phạm vi nghiên cứu: Phần mềm quản lý sản xuất nước uống tinh khiết đóng bình tại trường Đại học Sao Đỏ.

1. **Ý nghĩa thực tiễn của đề tài**

* Đơn giản hóa quá trình quản lý thao tác với dữ liệu.
* Tiết kiệm thời gian, công sức cho nhân viên và người quản lý.

1. **Bố cục của đề tài**

Nội dung của đồ án được trình bày trong 4 phần:

* Phần 1: Chương I: Cơ sở lý thuyết.
* Phần 2: Chương II: Phân tích và thiết kế hệ thống.
* Phần 3: Chương III: Xây dựng ứng dụng.
* Phần 4: Kết luận và hướng phát triển.

# CƠ SỞ LÝ THUYẾT

## 1.1 Giới thiệu về ngôn ngữ PHP

### 1.1.1 Giới thiệu chung về ngôn ngữ PHP

PHP - viết tắt của từ "Hypertext Preprocessor", là một ngôn ngữ lập trình kịch bản được chạy ở phía server nhằm sinh ra mã html trên client. Ngôn ngữ này được phát triển từ năm 1994 và cho đến nay đã được nhiều người sử dụng để phát triển các ứng dụng phần mềm thông qua lập trình web. Đặc điểm của ngôn ngữ lập trình này là sử dụng mã nguồn mở, dễ dàng nhúng vào HTML và tích hợp với web. PHP đã trải qua rất nhiều phiên bản và được tối ưu hóa cho các ứng dụng web, với cách viết mã rõ rãng, tốc độ nhanh, dễ học nên PHP đã trở thành một ngôn ngữ lập trình web rất phổ biến và được ưa chuộng.

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 và hệ điều hành Linux (LAMP).

* 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...) đóng vai trò là nơi lưu trữ và truy vấn dữ liệu.
* Linux: Hệ điều hành mã nguồn mở được sử dụng rất rộng rãi cho các webserver. Thông thường các phiên bản được sử dụng nhiều nhất là RedHat Enterprise Linux, Ubuntu...

### 1.1.2 Các tính năng của PHP

Các tính năng chính của PHP:

* Mã nguồn mở
* Nền tảng độc lập
* Sử dụng quy trình và hướng đến đối tượng
* Dễ học
* Phát triển ứng dụng và mục đích sử dụng
* Tương thích với máy chủ
* Tích hợp cơ sở dữ liệu dễ dàng
* Những ứng dụng được viết bằng ngôn ngữ PHP và được sử dụng để phát triển:
* Tạo ra các ứng dụng web như các trang web động các website thương mại điện tử
* Công cụ quản lý dự án code
* Tạo ứng dụng Facebook như Family Treen và eBuddy, hoặc cũng có thể được sử dụng để tạo ra các trang mạng xã hội.
* Tạo ra các ứng dụng trên điện thoại di động
* Các hệ thống quản lý nội dung như WordPress và Drupal
* Phát triển plugin WordPress
* Tạo tệp PDF

### 1.1.3 Phiên bản PHP 7

PHP7 là một phiên bản của ngôn ngữ PHP được giới thiệu vào ngày 3 tháng 9 năm 2015, đây là phiên bản mới nhất và được trông đợi nhất bởi có nhiều sự thay đổi về cú pháp, bổ sung các thư viện hiện đại giúp ngôn ngữ PHP trở nên đa dạng hơn. Để tránh sự nhầm lẫn giữa các version thì sau này chúng ta hay gọi PHP version 7 là PHP7, còn các version trước đó là PHP.

Đầu tiên phải kể đến đó là về hiệu năng, PHP 7 đã cải tiến hiệu năng vượt trội so với các phiên bản cũ. Ở các CMS hay các framework thì PHP 7 đều cho chúng ta một kết quả cực tốt. Cụ thể nó có thể nhanh gấp 2 lần so với các phiên bản cũ hơn. Và với các framework mạnh như Laravel và Zend cũng đều cho kết quả rất tốt.

Phiên bản mới đã cho phép chúng ta ràng buộc dữ liệu truyền vào hàm và ràng buộc dữ liệu trả về của hàm.

Toán tử null Coalesce: Giúp chúng ta gán giá trị hết sức ngắn gọn với toán tử ??

Toán tử Spaceship: Bổ sung toán tử <=>

Hằng có thể là mảng: Chúng ta có thể khai báo hằng là một mảng dữ liệu.

Lớp ẩn danh(anonymous classes): Cùng với hàm ẩn danh, hỗ trợ thêm lớp ẩn danh.

Nhóm use: Thay vì phải khai báo nhiều use cho các namespace cùng tên thì ta có thể gộp chúng lại một group use.

Hàm intdiv(): Bổ sung hàm intdiv() với chức năng lấy nguyên.

Hàm khởi tạo trùng tên class: Phiên bản này chỉ chấp nhận cách khai báo hàm khởi tạo bằng magicmethods.

Loại bỏ các extensions: ereg, mssql, mysql, sybase\_ct

Loại bỏ các SAPIs: aolserver, apache, apache\_hooks, apache2filter, caudium, continuity, isapi, milter, nsapi, phttpd, pi3web, roxen, thttpd, tux, webjames.

### 1.1.4 Làm việc với PHP bằng xampp

XAMPP là một phần mềm miễn phí với mã nguồn mở được phát triển bởi Apache. Gói phần mềm XAMPP chứa các bản phân phối Apache được sử dụng để xây dựng và phát triển các webssite và đặc biệt là ứng dụng web chạy bằng PHP.

Phần mềm máy chủ XAMPP cung cấp cho bạn môi trường thích hợp để thử nghiệm các dự án MYSQL, PHP, Apache và Perl trên máy tính cục bộ.

Dạng đầy đủ của XAMPP là X là viết tắt của Cross-platform, (A) máy chủ Apache, (M) MariaDB, (P) PHP và (P) Perl. Nền tảng của nó có thể chạy trên bất kỳ máy tính nào với bất kỳ hệ điều hành nào.

Cách cài đặt XAMPP rất đơn giản và nhanh chóng, khi XAMPP được cài đặt trên máy tính của bạn, nó hoạt động như một máy chủ cục bộ hoặc localhost. Bạn có thể kiểm tra các trang web trước khi tải nó lên máy chủ web từ xa. Phần mềm XAMPP này cung cấp cho bạn môi trường thích hợp để thử nghiệm các ứng dụng MYSQL, PHP, Apache và Perl trên máy tính.

## 1.2 Giới thiệu về cơ sở dữ liệu MySQL

### 1.2.1 Tổng quan về MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,…

Hệ thống quản lý cơ sở dữ liệu MySQL có nhiều ưu điểm như sau:

1. Tốc độ: MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà bạn có thể có.

2. Dễ sử dụng: MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn .

3. Giá thành: MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.

4. Hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Bạn cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity -một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).

5. Năng lực: Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn có thể truy cập MySQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web…

6. Kết nối và bảo mật: MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẽ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của bạn thì không thể nhìn được.

7. Tính linh động: MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.

8. Sự phân phối rộng: MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của bạn. Nếu bạn không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, bạn có thể lấy mã nguồn và tìm tòi nó. Nếu bạn không thích một vài cái, bạn có thể thay đổi nó.

9. Sự hỗ trợ: Bạn có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

### 1.2.2 Các phiên bản MySQL

MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

Với phiên bản 8.0, MySQL đã bỏ qua một số phiên bản của nó (từ bản 5,5), do bản 6.0 được không được chú ý và 7.0 được dành riêng cho các phiên bản nội bộ của MySQL.

Các tính năng được đưa ra trong MySQL 8.0 bao gồm:

* Thứ nhất hỗ trợ cho Unicode 9.0.
* Các tính năng cửa sổ và cú pháp SQL đệ quy, cho các truy vấn mà trước đây không thể hoặc khó thực hiện.
* Mở rộng hỗ trợ dữ liệu JSON và chức năng lưu trữ dữ liệu.

MySQL là một môi trường tiêu chuẩn đối với nhiều ứng dụng web.

### 1.2.3 Kiểu dữ liệu và thuộc tính

MySQL sử dụng nhiều kiểu dữ liệu, được chia thành 3 loại: kiểu số, kiểu date và time, và kiểu chuỗi.

Kiểu dữ liệu số trong MySQL

MySQL sử dụng tất cả các kiểu dữ liệu số theo chuẩn ANSI SQL, vì thế nếu bạn đã làm quen với một hệ thống cơ sở dữ liệu khác, thì bạn sẽ thấy những định nghĩa này là khá thân thuộc với bạn khi học về MySQL. Dưới đây liệt kê các kiểu dữ liệu số phổ biến và miêu tả của chúng:

* INT - Một số nguyên với kích cỡ thông thường, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -2147483648 tới 2147483647, nếu không dấu thì dãy giá trị là từ 0 tới 4294967295. Bạn có thể xác định một độ rộng lên tới 11 chữ số.
* TINYINT - Một số nguyên với kích cỡ rất nhỏ, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -128 tới 127, nếu không dấu thì dãy giá trị là từ 0 tới 255. Bạn có thể xác định một độ rộng lên tới 4 chữ số.
* SMALLINT - Một số nguyên với kích cỡ nhỏ, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -32768 tới 32767, nếu không dấu thì dãy giá trị là từ 0 tới 65535. Bạn có thể xác định một độ rộng lên tới 5 chữ số.
* MEDIUMINT - Một số nguyên với kích cỡ trung bình, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -8388608 tới 8388607, nếu không dấu thì dãy giá trị là từ 0 tới 16777215. Bạn có thể xác định một độ rộng lên tới 9 chữ số.
* BIGINT - Một số nguyên với kích cỡ lớn, có thể là signed hoặc unsigned. Nếu có dấu, thì dãy giá trị có thể là từ -9223372036854775808 tới 9223372036854775807, nếu không dấu thì dãy giá trị là từ 0 tới 18446744073709551615. Bạn có thể xác định một độ rộng lên tới 20 chữ số.
* FLOAT(M,D) - Một số thực dấu chấm động không dấu. Bạn có thể định nghĩa độ dài hiển thị (M) và số vị trí sau dấy phảy (D). Điều này là không bắt buộc và sẽ có mặc định là 10,2: với 2 là số vị trí sau dấu phảy và 10 là số chữ số (bao gồm các phần thập phân). Phần thập phân có thể lên tới 24 vị trí sau dấu phảy đối với một số FLOAT.
* DOUBLE(M,D) - Một số thực dấu chấm động không dấu. Bạn có thể định nghĩa độ dài hiển thị (M) và số vị trí sau dấy phảy (D). Điều này là không bắt buộc và sẽ có mặc định là 16,4: với 4 là số vị trí sau dấu phảy và 16 là số chữ số (bao gồm các phần thập phân). Phần thập phân có thể lên tới 53 vị trí sau dấu phảy đối với một số DOUBLE. REAL là đồng nghĩa với DOUBLE.
* DECIMAL(M,D) - Một kiểu khác của dấu chấm động không dấu. Mỗi chữ số thập phân chiếm 1 byte. Việc định nghĩa độ dài hiển thị (M) và số vị trí sau dấy phảy (D) là bắt buộc. NUMERIC là một từ đồng nghĩa cho DECIMAL.
* Kiểu dữ liệu Date và Time trong MySQL
* Kiểu dữ liệu Date và Time được phân loại thành:
* DATE - Một date trong định dạng YYYY-MM-DD, giữa 1000-01-01 và 9999-12-31. Ví dụ, ngày 25 tháng 12 năm 2015 sẽ được lưu ở dạng 2015-12-25.
* DATETIME - Một tổ hợp Date và Time trong định dạng YYYY-MM-DD HH:MM:SS, giữa 1000-01-01 00:00:00 và 9999-12-31 23:59:59. Ví dụ, 3:30 chiều ngày 25 tháng 12, năm 2015 sẽ được lưu ở dạng 2015-12-25 15:30:00.
* TIMESTAMP - Một Timestamp từ giữa nửa đêm ngày 1/1/1970 và 2037. Trông khá giống với định dạng DATETIME trước, khác biệt ở chỗ không có dấu gạch nối giữa các số. Ví dụ, 3:30 chiều ngày 25 tháng 12, năm 2015 sẽ được lưu dưới dạng 20151225153000 ( YYYYMMDDHHMMSS ).
* TIME - Lưu time trong định dạng HH:MM:SS.
* YEAR(M) - Lưu 1 năm trong định dạng 2 chữ số hoặc 4 chữ số. Nếu độ dài được xác định là 2 (ví dụ: YEAR(2)), YEAR có thể từ 1970 tới 2069 (70 tới 69). Nếu độ dài được xác định là 4, YEAR có thể từ 1901 tới 2155. Độ dài mặc định là 4.
* Kiểu dữ liệu chuỗi trong MySQL
* Ngoài các kiểu dữ liệu số hoặc kiểu dữ liệu date và time, thì còn một kiểu dữ liệu mà bạn thường sử dụng nhất trong MySQL là kiểu dữ liệu chuỗi. Dưới đây liệt kê các kiểu dữ liệu chuỗi và phần miêu tả của chúng trong MySQL:
* CHAR(M) - Một chuỗi có độ dài cố định có độ dài từ 1 tới 255 ký tự (ví dụ CHAR(5)). Nếu giá trị thật của một trường kiểu Char không bằng với độ dài khai báo thì phần thiếu bên phải của nó sẽ được thêm bằng các kí tự trắng một cách tự động. Định nghĩa độ dài là không bắt buộc, giá trị mặc định là 1.
* VARCHAR(M) - Dữ liệu kiểu chuỗi có độ dài thay đổi, có độ dài từ 1 đến 255 kí tự (ví dụ Varchar(24)). Bạn phải định nghĩa độ dài khi tạo một trường VARCHAR.
* BLOB hoặc TEXT - Trường kiểu này có độ dài tối đa 65535 kí tự. BLOBs là viết tắt của "Binary Large Objects", và được sử dụng để lưu trữ một lượng lớn dữ liệu nhị phân như các bức ảnh hoặc các loại tập tin khác. Với TEXT, trường cũng lưu trữ được một lượng lớn dữ liệu. Điểm khác nhau giữa chúng là: khi sắp xếp và so sánh dữ liệu đã lưu trữ thì với BLOBs là phân biệt kiểu chữ, còn với TEXT là không phân biệt kiểu chữ. Bạn không phải xác định độ dài với BLOBs hoặc TEXT.
* TINYBLOB hoặc TINYTEXT - Một cột BLOB hoặc TEXT với độ dài tối đa là 255 ký tự. Bạn không cần xác định độ dài với TINYBLOB hoặc TINYTEXT.
* MEDIUMBLOB hoặc MEDIUMTEXT - Một cột BLOB hoặc TEXT với độ dài tối đa là 16777215 ký tự. Bạn không cần xác định độ dài với MEDIUMBLOB hoặc MEDIUMTEXT.
* LONGBLOB hoặc LONGTEXT - Một cột BLOB hoặc TEXT với độ dài tối đa là 4294967295 ký tự. Bạn không cần xác định độ dài với LONGBLOB hoặc LONGTEXT.
* ENUM - Khi định nghĩa một trường kiểu này, tức là, ta đã chỉ ra một danh sách các đối tượng mà trường phải nhận (có thể là Null). Ví dụ, nếu ta muốn một trường nào đó chỉ nhận một trong các giá trị "A" hoặc "B" hoặc "C" thì ta phải định nghĩa kiểu ENUM cho nó như sau: ENUM ('A', 'B', 'C'). Và chỉ có các giá trị này (hoặc NULL) có thể xuất hiện trong trường đó.

### 1.2.4 Sử dụng MySQL bằng xampp

PhpMyAdmin là một công cụ mã nguồn mở miễn phí được viết bằng ngôn ngữ PHP nhằm giúp người dùng ( các nhà quản trị cơ sở dữ liệu …) có thể quản lý cơ sở dữ liệu MySQL thông qua giao diện web thay vì sử dụng giao diện cửa sổ dòng lệnh ( Command line interface). Sử dụng phpMyAdmin người dùng có thể thực hiện nhiều tác vụ khác nhau như khi sử dụng cửa sổ dòng lệnh. Bao gồm như việc tạo, cập nhật và xóa cơ sở dữ liệu, các bảng, phân quyền user ….

Các tính năng chính:

1. Giao diện web

2. Quản lý cơ sở dữ liệu MySQL

3. Nhập dữ liệu từ CSV và SQL

4. Xuất dữ liệu sang các định dạng khác nhau: CSV, SQL, XML, PDF (thông qua thư viện TCPDF), ISO/IEC 26300 – OpenDocument văn bản và bảng tính, Word, Excel, LaTeX và các định dạng khác

5. Quản lý nhiều máy chủ

6. Tạo PDF đồ họa của bố trí cơ sở dữ liệu

7. Tạo các truy vấn phức tạp bằng cách sử dụng Query-by-example (QBE)

8. Tìm kiếm tổng quan trong cơ sở dữ liệu hoặc một tập hợp con của nó

9. Chuyển đổi dữ liệu được lưu trữ thành các định dạng bằng cách sử dụng một tập hợp các chức năng được xác định trước, như hiển thị dữ liệu BLOB như hình ảnh hoặc tải về liên kết

10. Giám sát các truy vấn

PHPMyAdmin được tích hợp sẵn vào Xampp trong khi cài đặt, nó giúp cho việc thực hiện các công việc như xem danh sách các database, cấu trúc table, chèn dữ liệu và thay đổi cấu trúc bảng một cách nhanh chóng và trực quan.

## 1.3 Giới thiệu về kiến trúc MVC

### 1.3.1 Tổng quan về kiến trúc MVC

Mô hình MVC là một trong những cái tên không thể nào không biết khi tiếp cận trong việc lập trình web.

MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Mô hình MVC và các thành phần bên trong của MVC:

Model (M):

Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…

View (V):

Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.

Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.

Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong Controller.

**Controller (C):**

Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

**Luồng tương tác giữa các thành phần trong MVC:**

Mỗi bộ phận thực hiện chức năng nhất định, nhưng chúng có sự thống nhất, liên kết với nhau tạo nên mô hình MVC. Mô hình này tương đối nhẹ. Nó có thể tích hợp được nhiều tính năng có trong ASP.NET hiện giờ.

**Luồng đi trong mô hình MVC**:

Có thể hình dung, khi một yêu cầu từ máy client gửi tới server, Controller sẽ thực hiện nhiệm vụ của mình đó là tiếp nhận và xử lý yêu cầu. Trong trường hợp cần thiết, nó có thể liên hệ Model – bộ phận làm việc với database để hỗ trợ.

Khi xử lý xong yêu cầu, kết quả sẽ được trả về View. Tại View sẽ tạo mã HTML thành giao diện và trả về hiển thị trên trình duyệt.

### 1.3.2 Ưu điểm của kiến trúc MVC

Nhẹ, tiết kiệm băng thông: MVC không sử dụng viewstate nên khá tiết kiệm diện tích băng thông. Khi sử dụng, người dùng có thể sử dụng ứng dụng trên web cần tương tác gửi và nhận dữ liệu một cách liên tục. Do đó, việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn.

Ưu điểm của mô hình MVC.

* Kiểm tra dễ dàng: Với MVC, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.
* Chức năng control: Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, Javascript có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.
* View và size: View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.
* Chức năng Soc (Separation of Concern): Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.
* Tính kết hợp: Việc tích hợp ở mô hình MVC cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.
* Đơn giản: Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.

### 1.3.3 Framewok CodeIgniter

CodeIgniter là một nền tảng ứng dụng web (web application framework) nguồn mở được dùng để xây dựng các ứng dụng web động tương tác với PHP. Nó cho phép các nhà phát triển xây dựng một ứng dụng web nhanh hơn - so với việc viết mã hỗn tạp - bằng cách cung cấp 1 bộ thư viện đầy đủ cho các tác vụ thông thường, cũng như cung cấp một mô hình tương tác đơn giản và dễ hiểu cho việc kết nối tới những bộ thư viện đó. Phiên bản chính thức đầu tiên của CodeIgniter được công bố vào 28 tháng 2 năm 2006.

Những ưu điểm của Codeigniter Framework

* Framework Codeigniter sở hữu các ưu điểm đặc biệt khiến nhiều lập trình viên, nhà phát triển web đã lựa chọn trong quá trình thiết kế web của mình. Các ưu điểm của Codeigniter Framework có thể kể đến như sau:
* Footprint nhỏ: Dung lượng CodeIgniter 3 chỉ tốn 2MB khi download, bao gồm cả file cài đặt lẫn hướng dẫn sử dụng.
* Hướng dẫn sử dụng rõ ràng, dễ hiểu: Hướng dẫn sử dụng CodeIgniter được đính kèm trong file tải xuống. Nó bao gồm phần giới thiệu, hướng dẫn, một số hướng dẫn “Làm thế nào để…”, và tài liệu tham khảo cho các lập trình viên hoàn thiện website của mình.
* Khả năng tương thích với lưu trữ chuẩn: CodeIgniter 3 chỉ cần PHP 5.3.7, và có thể hoạt động tương thích với hầu hết các hosting chia sẻ hoặc hosting riêng. Nhiều ứng dụng web cần một cơ sở dữ liệu, và CodeIgniter cũng hỗ trợ phổ biến nhất, bao gồm cả MySQL.
* Được thiết kế theo mô hình Model-View-Controller: Mô hình MVC giúp tách thành phần hiển thị giao diện và xử lý của một phần mềm thành các phần độc lập, từ đó giúp cho việc thiết kế, xử lý và bảo trì mã nguồn dễ dàng hơn.
* Hệ thống thư viện đa dạng: CodeIgniter cung cấp các thư viện phục vụ cho những tác vụ thường gặp nhất trong lập trình web, chẳng hạn như truy cập cơ sở dữ liệu, gửi email, kiểm tra dữ liệu, xử lí hình ảnh…
* Cấu hình gần như bằng không: Phần lớn cấu hình CodeIgniter được thực hiện theo quy ước, ví dụ như đặt mô hình trong một “mô hình” thư mục. Bên cạnh đó còn có một số tùy chọn cấu hình có sẵn, thông qua các tập lệnh trong thư mục “config”.
* Tốc độ nhanh: CodeIgniter được đánh giá là một PHP framework có tốc độ nhanh chóng. Một server bình thường có thể đáp ứng được hàng triệu truy cập / ngày.
* Miễn phí: CodeIgniter được phát hành dưới giấy phép Apache/BSD mở rộng, cho phép người dùng tự do thay đổi, phát triển và phân phối mã nguồn mở này theo sự sáng tạo của bản thân.
* Với sự hỗ trợ từ Framewwork Codeigniter, nhiều website đã đạt được tiêu chuẩn MVC và hoạt động tốt giữa hàng ngàn website có mặt trên Internet hiện nay.
* Hỗ trợ SEO: Cấu trúc URL của CodeIgniter rất thân thiện với các công cụ tìm kiếm. Điều này nghiễm nhiên giúp cho website của bạn tăng thứ hạng nhanh hơn mà không cần phải đầu tư quá nhiều phần mềm hay công cụ.
* Bảo mật hệ thống: Cơ chế kiểm tra dữ liệu chặt chẽ, ngăn ngừa XSS và SQL Injection của CodeIgniter giúp giảm thiểu các nguy cơ xâm nhập, làm hại đến hệ thống

### 1.3.4 Bootstrap

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 webiste 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.

Ưu điểm khi sử dụng boostrap

* 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 đuổi. CDN Boostrap 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 trang website theo khung browser. Mục đích để phù hợp với màn hình của máy tính để bàn, tablet hay laptop.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Khảo sát hệ thống

### 2.1.1 Hệ thống cũ

Xưởng sản xuất nước tinh khiết đóng bình đặt tại cơ sở 2 – trường Đại học Sao Đỏ. Xưởng chuyên sản xuất, cung cấp các sản phẩm nước tinh khiết đóng bình như: bình nước uống 20l, nước tinh khiết đóng chai loại 500ml, nước tinh khiết đóng chai loại 350ml.

Xưởng nhập các nguyên vật liệu như bình, chai đựng nước, tem, nhãn từ nhà cung cấp quen thuộc, tin cậy. Sử dụng các thiết bị sản xuất hiện đại, trải qua các quá trình như: lọc thô, lọc tinh, lọc micro, khử trùng bằng tia UV,.., Các sản phẩm được sản xuất đều đạt chất lượng và độ an toàn cao cho người sử dụng.

Sản phẩm của xưởng phục vụ chủ yếu cho là các phòng, ban, khu vực trong trường Đại học Sao Đỏ và các hộ kinh doanh gần khu vực trường.

Sau khi phân tích xu hướng, nhu cầu của khách hàng và số lượng tồn kho nguyên vật liệu, người quản lý sẽ liên hệ với nhà cung cấp để nhập thêm lượng nguyên vật liệu phù hợp.

Trước khi nguyên vật liệu được nhập vào kho thì người quản lý cần phải làm các công việc như sau:

+ Nhập thông tin của nguyên vật liệu, số lượng, giá bán

+ Kiểm tra và nhận hàng

+ Viết phiếu nhập hàng và lưu trữ lại để thống kê, báo cáo

**a. Mô tả quy trình sản xuất và tiêu thụ**

- Lượng nguyên vật liệu được nhập về luôn được duy trì cân bằng với lượng tiêu thụ sản phẩm, ngoài ra xưởng còn lưu trữ một lượng nguyên vật liệu nhất định để dự phòng.

- Sản phẩm được sản xuất qua các giai đoạn: lọc thô, lọc tinh, lọc mico, khử trùng bằng UV, đóng chai, dán tem – nhãn. Các ca sản xuất hoạt động với chu kỳ theo tuần vào mùa lạnh và khoảng 3-4 ngày vào mùa nóng khi nhu cầu nước uống tăng cao.

- Sản phẩm được tập hợp tại kho. Khi có đơn đặt hàng thì sẽ được chuyển lên xe vận chuyển đến tận nơi cho khách hàng.

- Lượng sản phẩm bán ra sẽ được ghi lại để thống kê.

**b. Thống kê và báo cáo**

- Việc thống kê và báo cáo được thực hiện cuối mỗi tháng và mỗi quý. Báo cáo sẽ được gửi về trường đại học Sao Đỏ.

C, Một số biểu mẫu

### 2.1.2 Đề xuất xây dựng hệ thống mới

Trên cơ sở các phân tích thực trạng của hệ thống quản lý sản xuất hiện tại, để thực hiện các khâu nhập hàng, ghi chép sản xuất, hoá đơn bán hàng và thống kê hiệu quả hơn, hệ thống quản lý sản xuất nước uống đóng bình được xây dựng với các chức năng sau:

- Chức năng hệ thống: cho phép người dùng đăng nhập vào hệ thống, thao tác với tài khoản

- Chức năng quản lý sản phẩm: thêm, sửa, xoá các loại sản phẩm, sản phẩm cũng như số lượng các sản phẩm có trong kho

- Chức năng quản lý nhập hàng: thêm, sửa, xoá hoá đơn nhập

- Chức năng quản lý sản xuất: thêm, sửa, xoá hoá đơn sản xuất gồm lượng nguyên vật liệu tiêu hao và lượng thành phẩm tạo ra.

- Chức năng quản lý khách hàng: thêm, sửa, xoá thông tin khách hàng, quản lý công nợ, số vỏ bình mượn

- Chức năng thống kê: thống kê theo thời gian hoá đơn nhập, xuất hay sản xuất, số lượng, tổng tiền các loại sản phẩm.

## 2.2 Phân tích xây dựng hệ thống mới

### 2.2.1 Xác định các tác nhân

Sau khi khảo sát hiện trạng thu được thông tin sau:

- Quản lý sản phẩm: mỗi sản phẩm được quản lý bởi các thông tin sau đây: mã sản phẩm, tên sản phẩm, mã loại sản phẩm, giá, số lượng, mô tả

- Quản lý tài khoản: mỗi tài khoản quản lý bởi mã nhân viên, tên đăng nhập, mật khẩu, email, loại tài khoản

- Quản lý hoá đơn: mỗi hoá đơn quản lý bởi mã hoá đơn, tên khách hàng, điện thoại, địa chỉ, ngày lập hoá đơn, tổng tiền, chi tiết hoá đơn.

- Quản lý sản xuất: mỗi đơn sản xuất quản lý bởi mã đơn sản xuất, ngày lập đơn, chi tiết đơn

- Quản lý nhập hàng: mỗi đơn nhập hàng được quản lý bởi mã đơn nhập hàng, tên người nhập, ngày lập đơn.

Từ quy trình đã nêu, ta nhận thấy rằng phần mềm được xây dựng cho bài toán đặt ra chủ yếu phục vụ cho đối tượng người quản lý xưởng sản xuất. Người quản lý sau khi đăng nhập sẽ được sử dụng toàn bộ chức năng trong phần mềm.

### 2.2.2 Đặc tả yêu cầu

Người quản lý: là người làm chủ phần mềm, có quyền sử dụng mọi chức năng của phẩn mềm. Người quản lý được cấp một tên đăng nhập và mật khẩu để đăng nhập vào hệ thống và thực hiện những chức năng có trong hệ thống. Nếu quá trình đăng nhập thành công thì người quản lý có những chức năng sau:

- Quản lý sản phẩm

- Quản lý loại sản phẩm

- Quản lý tài khoản

- Quản lý hoá đơn bán hàng

- Quản lý đơn nhập hàng

- Quản lý đơn sản xuât

- Quản lý khách hàng

- Thống kê

Do xưởng sản xuất không lắp đặt máy tính và việc hoàn thành hoá đơn có thể phải di chuyển nhiều nơi, chúng em đã quyết định chuyển nền tảng xây dựng phần mềm từ Csharp sang web PHP vì:

- Nếu dùng C#:

+ Cần lắp đặt thêm máy tính tại xưởng, làm tăng chi phí

+ Để dùng được phần mềm cần sử dụng máy tính với cấu hình tương đối: cài Windows 10, .NET Framework 4.7, SQL Server,..

+ Việc nhập liệu chỉ có thể thực hiện tại một nơi cố định

- Lợi ích khi chuyển sang PHP:

+ PHP miễn phí, không yêu cầu bản quyền.

+ Kết hợp với MySQL có chức năng cần dùng tương đương SQL Server

+ Xây dựng trên nền tảng web nên có thể truy cập ở mọi lúc, mọi nơi có internet.

+ Không yêu cầu nhiều về phần cứng, chỉ cần máy tính hoặc điện thoại di động có kết nối internet ổn định là có thể sử dụng.

## 2.3 Xây dựng các biểu đồ

### 2.3.1 Biểu đồ phân cấp chức năng



Hình . Biểu đồ phân cấp chức năng

Khi người quản lý đăng nhập vào hệ thống sẽ được sử dụng toàn bộ các chức năng sau: quản lý tài khoản, quản lý sản phẩm, quản lý loại sản phẩm, quản lý hoá đơn, quản lý nhập hàng, quản lý sản xuất, thống kê.

- Chức năng quản lý tài khoản:

+ Đăng nhập vào hệ thống với tên đăng nhập và mật khẩu

+ Thêm, sửa, xoá tài khoản

+ Đăng xuất

- Chức năng quản lý sản phẩm: thêm, sửa, xoá sản phẩm, quản lý số lượng.

- Chức năng quản lý loại sản phẩm: thêm, sửa, xoá loại sản phẩm.

- Chức năng quản lý hoá đơn: thêm, sửa, xoá hoá đơn, tự động tính tiền hoá đơn và cập nhật kho khi thêm hoá đơn mới.

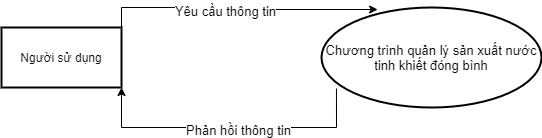
- Chức năng quản lý nhập hàng: thêm, sửa, xoá đơn nhập hàng, cập nhật kho

- Chức năng quản lý sản xuất: thêm, sửa, xoá đơn sản xuất, cập nhật kho

- Chức năng quản lý khách hàng: thêm, sửa, xoá thông tin khách hàng, cập nhật công nợ và số bình mượn

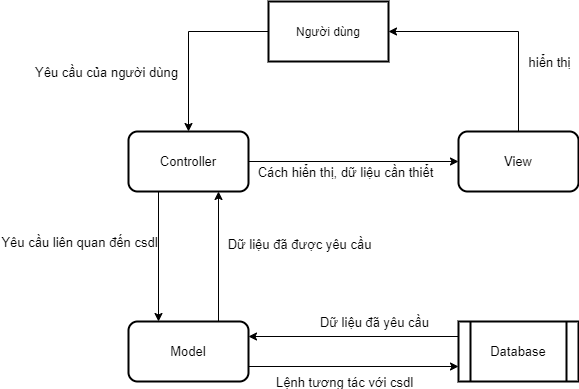
- Chức năng thống kê: thống kê hoá đơn, đơn nhập hàng, đơn sản xuất theo ngày, thống kê tổng tiền các hoá đơn, thống kê số lượng các loại sản phẩm, nguyên vật liệu.

### 2.3.2 Biểu đồ luồng dữ liệu



Hình . Biểu đồ luồng dữ liệu mức khung cảnh

Khi người quản lý yêu cầu thông tin thì hệ thống sẽ trả về kết quả yêu cầu.



Hình . Biểu đồ luồng dữ liệu mức đỉnh

Khi người dùng truy cập vào hệ thống, người dùng sẽ tương tác với controller. Controller sẽ tiếp nhận yêu cầu từ người dùng, phân tích và điều hướng yêu cầu đó. Nếu yêu cầu không liên quan đến cơ sở dữ liệu, controller sẽ chỉ gọi đến view, load view cho người dùng. Nếu yêu cầu cần tác động vào cơ sở dữ liệu, yêu cầu sẽ được điều hướng đến model, đây là thành phần có chức năng làm việc với cơ sở dữ liệu. Sau khi model làm việc với cơ sở dữ liệu, kết quả trả về sẽ chuyển trở lại controller. Ở đây, dữ liệu sẽ được kết hợp với view để hiển thị cho người dùng.

## 2.4 Phân tích thiết kế cơ sở dữ liệu

### 2.4.1 Xác định các thực thể

SANPHAM (sản phẩm): id(mã hàng), tensanpham(tên sản phẩm), maloaisp (mã loại sản phẩm), gia(đơn giá), soluong(số lượng), mota(mô tả).

LOAISANPHAM (loại sản phẩm): id(mã loại sản phẩm), tenloai(tên loại sản phẩm)

USER(tài khoản): id(mã tài khoản), username(tên đăng nhập), password(mật khẩu), email(thư điện tử), level(loại tài khoản).

HOADON(hoá đơn): id(mã hoá đơn), idkhachhang(mã khách hàng), ngaylap(ngày lập đơn), phivanchuyen(phí vận chuyển), tongtien(tổng tiền hoá đơn), thanhtoan(trạng thái thanh toán), ghichu(ghi chú).

CHITIETHOADON(chi tiết hoá đơn): id(mã chi tiết hoá đơn), mahoadon(mã hoá đơn), masanpham(mã sản phẩm), soluong(số lượng), gia(đơn giá).

NHAPHANG(nhập hàng): id(mã nhập hàng), tennguoinhap(tên nhập hàng), ngaylap(ngày lập đơn), phivanchuyen(phí vận chuyển), tongtien(tổng tiền nhập hàng), ghichu(ghi chú).

CHITIETNHAPHANG(chi tiết nhập hàng): id(mã chi tiết nhập hàng), manhaphang(mã nhập hàng), masanpham(mã sản phẩm), soluong(số lượng), gia(đơn giá).

SANXUAT(sản xuất): id(mã đơn sản xuất), ngaylap(ngày lập đơn), ghichu(ghi chú)

CHITIETSANXUAT(chi tiết sản xuất): id(mã chi tiết đơn sản xuất), masanxuat(mã đơn sản xuất), masanpham(mã sản phẩm), maloaisp(mã loại sản phẩm), soluong(số lượng).

KHACHHANG(khách hàng): id(mã khách hàng), tenkhachhang(tên khách hàng), dienthoai(số điện thoại), diachi(địa chỉ), congno(công nợ), vobinh(số vỏ bình mượn), ghichu(ghi chú).

### 2.4.2 Mô tả chi tiết các thực thể

Bảng . Thực thể USER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã người dùng |
| username | varchar | 100 | unique | Tên đăng nhập |
| password | char | 32 |  | Mật khẩu |
| email | varchar | 100 |  | Email |
| level | int | 1 |  | Loại tài khoản |

Bảng . Thực thể KHACHHANG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã khách hàng |
| tenkhachhang | text |  |  | Tên khách hàng |
| dienthoai | text |  |  | Số điện thoại |
| diachi | text |  |  | Địa chỉ |
| congno | int | 11 |  | Công nợ |
| vobinh | int | 11 |  | Số vỏ bình mượn |
| ghichu | text |  |  | Ghi chú |

Bảng . Thực thể SANPHAM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã sản phẩm |
| tensanpham | text |  |  | Tên sản phẩm |
| maloaisp | int | 11 | Foreign key | Mã loại sản phẩm |
| gia | int | 11 |  | Giá bán |
| soluong | int | 11 |  | Số lượng trong kho |
| mota | text |  |  | Mô tả |

Bảng . Thực thể LOAISANPHAM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã loại sản phẩm |
| tenloai | text |  |  | Tên loại sản phẩm |

Bảng . Thực thể HOADON

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã hoá đơn |
| idkhachhang | int | 11 | Foreign key | Mã khách hàng |
| ngaylap | date |  |  | Ngày lập đơn |
| phivanchuyen | int | 11 |  | Phí vận chuyển |
| tongtien | int | 11 |  | Tổng tiền hoá đơn |
| thanhtoan | int | 11 |  | Trạng thái thanh toán |
| ghichu | text |  |  | Ghi chú |

Bảng . Thực thể CHITIETHOADON

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã chi tiết hoá đơn |
| mahoadon | int | 11 | Foreign key | Mã hoá đơn |
| masanpham | int | 11 | Foreign key | Mã sản phẩm |
| soluong | int | 11 |  | Số lượng |
| gia | int | 11 |  | Giá bán |

Bảng . Thực thể NHAPHANG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã đơn nhập |
| tennguoinhap | text |  |  | Tên người nhập hàng |
| nhacungcap | text |  |  | Đơn vị cung cấp |
| ngaylap | date |  |  | Ngày lập đơn |
| phivanchuyen | int | 11 |  | Phí vận chuyển |
| tongtien | int | 11 |  | Tổng tiền đơn hàng |
| ghichu | text |  |  | Ghi chú |

Bảng . Thực thể CHITIETNHAPHANG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã chi tiết đơn nhập |
| manhaphang | int | 11 | Foreign key | Mã đơn nhập |
| masanpham | int | 11 | Foreign key | Mã sản phẩm |
| soluong | int | 11 |  | Số lượng |
| gia | int | 11 |  | Giá bán |

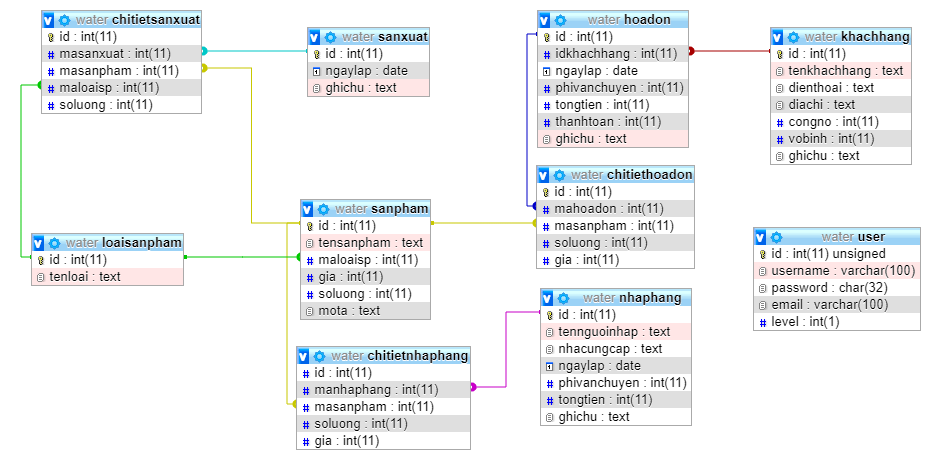
Bảng . Thực thể SANXUAT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã đơn sản xuất |
| ngaylap | date |  |  | Ngày lập đơn |
| ghichu | text |  |  | Ghi chú |

Bảng . Thực thể CHITIETSANXUAT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Độ rộng** | **Ràng buộc** | **Diễn giải** |
| id | int | 11 | Primary key,  Auto increment | Mã chi tiết đơn sản xuất |
| masanxuat | int | 11 | Foreign key | Mã đơn sản xuất |
| masanpham | int | 11 | Foreign key | Mã sản phẩm |
| maloaisanpham | int | 11 | Foreign key | Mã loại sản phẩm |
| soluong | int | 11 |  | Số lượng |

### 2.4.3 Sơ đồ liên kết các thực thể



Hình . Sơ đồ liên kết các thực thể

## 2.5 Áp dụng mô hình MVC và ưu điểm

### 2.5.1 Mô hình MVC trong xây dựng chương trình

Để ví dụ cho việc áp dụng mô hình MVC vào xây dựng phần mềm, ta sẽ trình bày việc xây dựng chức năng quản lý khách hàng trong chương trình.

a, Model

Sau khi đã tạo bảng khachhang trong cơ sở dữ liệu MySQL, ta tiến hành tạo file Khachhang\_Model.php trong thư mục models để tương tác với bảng này.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Khachhang\_Model extends CI\_Model

{

protected $table = 'khachhang';

public function \_\_construct()

{

parent::\_\_construct();

//Do your magic here

}

}

Lớp Khachhang\_Model kế thừa lớp CI\_Model có sẵn trong Framework để có thể sử dụng những hàm được xây dựng sẵn.

Thuộc tính $table được thiết lập để các hàm khi sử dụng chỉ cần sử dụng biến này thay cho việc phải viết ‘khachhang’ khi cần dùng. Khi có sự thay đổi về tên bảng tương tác, ta chỉ cần thay đổi thuộc tính $table từ đầu mà không cần phải thay đổi tên bảng trong tất cả các hàm trong lớp.

Hàm \_\_construct() là hàm khởi tạo lớp, sẽ gọi đến các hàm khởi tạo trong lớp cha và thiết lập những lớp liên quan cần dùng. Sau đó chúng ta cần bổ sung các hàm để tương tác với bảng khachhang.

public function getList()

{

$this->db->order\_by('tenkhachhang', 'ASC');

return $this->db->get($this->table)->result\_array();

}

public function getKhachhangById($id)

{

$this->db->where("id", $id);

return $this->db->get($this->table)->row\_array();

}

Đầu tiên là lấy dữ liệu. Hàm getList() sẽ lấy toàn bộ thông tin trong bảng và trả về kết quả được sắp xếp theo tên khách hàng từ a-z. Hàm getKhachhangById($id) sẽ lấy toàn bộ thông tin của một khách hàng dựa theo mã khách hàng được truyền vào.

Tiếp theo là thêm, sửa, xoá:

public function deleteKhachhang($id)

{

$this->db->where("id", $id);

$this->db->delete($this->table);

}

public function insertKhachhang($data\_insert)

{

$this->db->insert($this->table, $data\_insert);

}

public function updateKhachhang($data\_update, $id)

{

$this->db->where("id", $id);

$this->db->update($this->table, $data\_update);

}

Hàm insertKhachhang() dùng để thêm khách hàng. Dữ liệu truyền vào sẽ được Controller đóng gói thành một biến $data\_insert sau đó truyền kèm theo khi gọi lệnh. Tương tự với hàm updateKhachhang() để cập nhật khách hàng và deleteKhachhang() để xoá khách hàng.

b, View

Để dễ dàng quản lý các thành phần giao diện người dùng, các thành phần của view được chia nhỏ thành nhiều file. Trong folder views, tạo folder khachhang để chứa các thành phần view của chức năng quản lý khách hàng. Trong folder khachhang vừa tạo, tiếp tục tạo 4 file main.php, index\_view.php, add\_view.php, edit\_view.php.

File main.php là khung sườn chính để bố trí các thành phần trong view.

<!doctype html>

<html lang="en">

<?php $this->load->view("fontend/status"); ?>

<head>

<title><?php echo $title; ?></title>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

</head>

<body>

<div class="container-fluid">

<?php $this->load->view("fontend/header"); ?>

<?php $this->load->view($subview); ?>

<?php $this->load->view("fontend/footer"); ?>

</div>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</body>

</html>

Đầu tiên, sẽ gọi file status.php trong thư mục fontend để kiểm tra tình trang đăng nhập. Nếu chưa đăng nhập sẽ bị chuyển về trang đăng nhập. Trong thẻ <head> là nơi khai báo thuộc tính trang và css. Trong thẻ <body> là cách bố trí trang, load các header, footer từ folder fontend, khai báo scripts. Dùng lệnh <?php $this->load->view($subview); ?> để load view mà mình muốn hiển thị, đó là một trong 3 file index\_view.php, add\_view.php, edit\_view.php.

Tạo file index\_view.php:

<a name="" id="" class="btn btn-success" href="<?php echo base\_url(); ?>khachhang/add" role="button">Thêm khách hàng</a>

<table class="table table-striped table-inverse table-responsive">

<thead class="thead-inverse">

<tr>

<th>STT</th><th>Mã khách hàng</th>

<th>Tên khách hàng</th><th>Số điện thoại</th>

<th>Địa chỉ</th><th>Công nợ</th>

<th>Vỏ bình mượn</th><th>Ghi chú</th>

<th>Chỉnh sửa</th><th>Xoá</th>

</tr>

<?php $stt = 0;

foreach ($info as $item) { $stt++; ?>

<tr>

<td><?php echo $stt; ?></td>

<td><?php echo $item['id']; ?></td>

<td><?php echo $item['tenkhachhang']; ?></td>

<td><?php echo $item['dienthoai']; ?></td>

<td><?php echo $item['diachi']; ?></td>

<td><?php echo $item['congno']; ?></td>

<td><?php echo $item['vobinh']; ?></td>

<td><?php echo $item['ghichu']; ?></td>

<td><a name="" id="" class="btn btn-warning" href="<?php echo base\_url() . 'khachhang/edit/' . $item['id'] ?>" role="button">Sửa</a></td>

<td><a name="" id="" class="btn btn-danger" href="<?php echo base\_url() . 'khachhang/del/' . $item['id'] ?>" role="button">Xoá</a></td>

</tr>

<?php } ?>

</tr>

<tr>

<td colspan="9" align="center">Có tổng cộng <?php echo $total\_khachhang; ?> khách hàng.<br /></td>

</tr>

</table>

Do trong file main.php chúng ta đã khai báo css, scripts và các thẻ html cần thiết vì vậy ta không cần phải khai báo lại trong file này. Khi index\_view.php được load vào main.php thì các thẻ sẽ được hiển thị như file html thông thường. Tương tự với 2 file còn lại.

c, Controller

Sau khi đã hoàn thành Khachhang\_Model ta sẽ xây dựng Controller để điều khiển chức năng quản lý khách hàng. Tạo file Khachhang.php trong folder controllers.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Khachhang extends CI\_Controller

{

protected $\_data;

public function \_\_construct()

{

parent::\_\_construct();

//Do your magic here

$this->load->model('Khachhang\_Model', 'KH\_Model');

}

}

Lớp Khachhang kế thừa lớp CI\_Controller được xây dựng sẵn trong framework. Thuộc tính $\_data để đóng gói dữ liệu và truyền lại giữa controller với view. Hàm khởi tạo sẽ có thêm câu lệnh load model mà ta đã tạo ở phẩn a. Thêm hàm index(), đây là hàm được thực hiện mặc định mỗi khi controller được gọi.

public function index()

{

$this->\_data['subview'] = 'khachhang/index\_view.php';

$this->\_data['title'] = 'Danh sách Khách hàng';

$this->\_data['info'] = $this->KH\_Model->getList();

$this->\_data['total\_khachhang'] = $this->KH\_Model->countAll();

$this->\_data['mess'] = $this->session->flashdata('flash\_mess');

$this->load->view('khachhang/main.php', $this->\_data);

}

Chúng ta sẽ lần lượt truyền các thông tin vào biến $\_data truyền sang view. Tương tự, thêm các hàm add(), edit(), delete($id) để làm chức năng thêm, sửa, xoá khách hàng.

### 2.5.2 Ưu điểm khi sử dụng mô hình MVC

- Phân chia các thành phần riêng biệt: các thành phần với chức năng khác nhau được phân tách một cách rõ ràng, dễ dàng hơn cho việc xây dựng và phát triển: model chỉ làm việc với database, view chỉ hiển thị cho người dùng, controller điều khiển chức năng và dữ liệu.

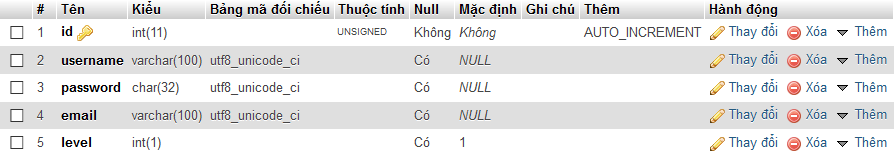
- Phát triển đơn giản hơn: thay vì phải viết hỗn hợp php và html tổng hợp cho cả việc hiển thị, xử lý dữ liệu, liên kết giữa các trang thì ta tách chúng ra làm nhiều phần đơn giản hơn. Các thành phần sẽ nhỏ hơn, dễ dàng xây dựng và kiểm soát lỗi hơn, chỉ được gọi khi cần thiết để tiết kiệm tài nguyên.

- Dễ dàng nâng cấp, bảo trì: các thành phần được chia nhỏ giúp dễ nâng cấp và bảo trì hơn. Ví dụ: với folder fontend, ta lưu các file chứa header, footer của trang web. Khi thay đổi mã ở đây thì giao diện trang web ở tất cả các trang sẽ được thay đổi. Nếu sử dụng phương pháp truyền thống ta sẽ phải thay đổi lần lượt tất cả các file trong hệ thống để đạt được hiệu quả tương tự. Nếu sử dụng template thì cũng có nguy cơ xảy ra lỗi liên quan đến đường dẫn.

# XÂY DỰNG CHƯƠNG TRÌNH

## 3.1 Thiết kế cơ sở dữ liệu với MySQL

### 3.1.1 Bảng quản lý tài khản (user)



Hình . Bảng user

### 3.1.2 Bảng loại sản phẩm (loaisanpham)



Hình . Bảng loaisanpham

### 3.1.3 Bảng sản phầm (sanpham)



Hình . Bảng sanpham

### 3.1.4 Bảng hoá đơn (hoadon)



Hình . Bảng hoadon

### 3.1.5 Bảng chi tiết hoá đơn (chitiethoadon)



Hình . Bảng chitiethoadon

### 3.1.6 Bảng nhập hàng (nhaphang)



Hình . Bảng nhaphang

### 3.1.7 Bảng chi tiết nhập hàng (chitietnhaphang)



Hình . Bảng chitietnhaphang

### 3.1.8 Bảng sản xuất (sanxuat)



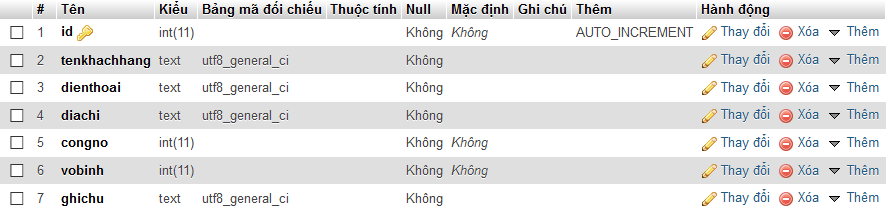
Hình . Bảng sanxuat

### 3.1.9 Bảng chi tiết sản xuất (chitietsanxuat)



Hình . Bảng chitietsanxuat

### 3.1.10 Bảng khách hàng

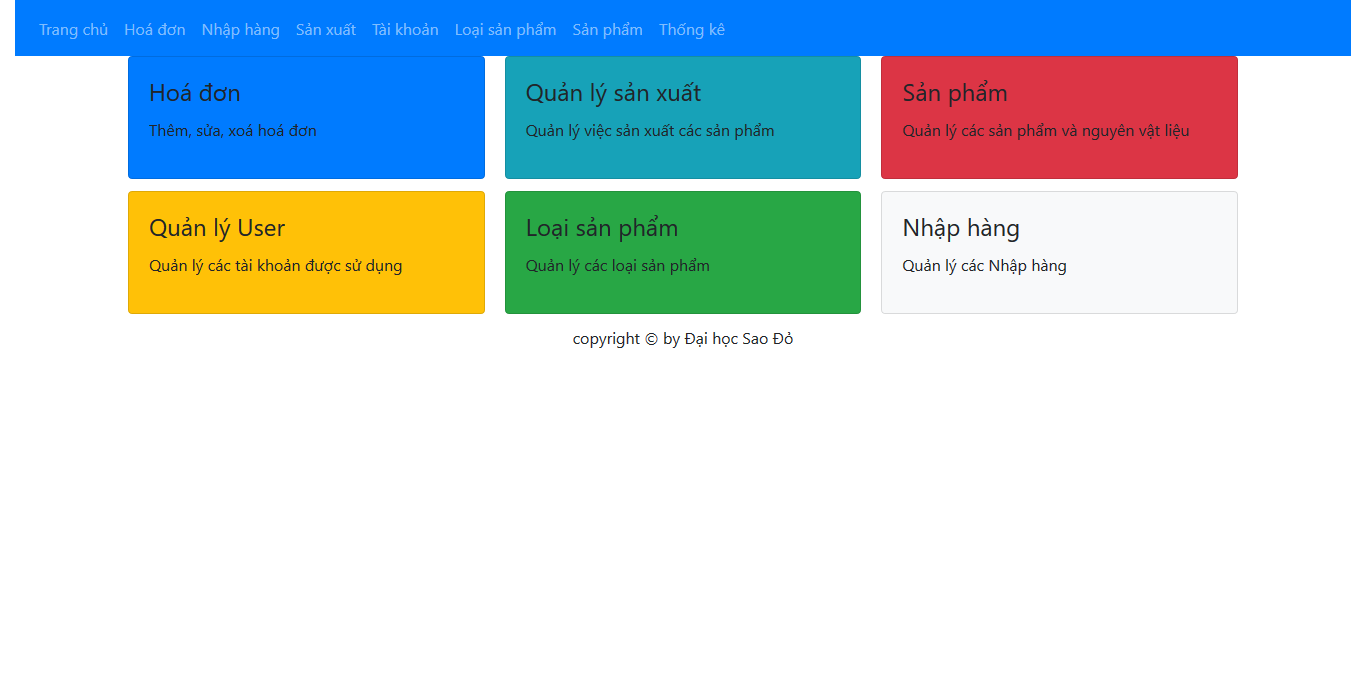


Hình . Bảng khachhang

## 3.2 Thiết kế chức năng

### 3.2.1 Trang chủ

**A, View**



Hình . Giao diện trang chủ trên máy tính



Hình . Giao diện trang chủ trên màn hình điện thoại

Tạo file index.php trong folder view.

Phần body của trang index.php sẽ được tự động tải nội dung của 3 file header.php, content.php, footer.php nằm trong thư mục fontend. Việc chia nhỏ file index.php thành nhiều file con giúp chúng ta có khả năng hạn chế lỗi khi file quá lớn, đồng thời có thể tận dụng lại những file đã viết một cách dễ dàng để áp dụng vào những chức năng khác mà không phải viết lại từ đầu. Khi muốn thay đổi một phần hoặc xảy ra lỗi ở file nào đó, ta chỉ cần sửa lại một phần nhỏ mã ở file đó thì việc sửa chữa sẽ được áp dụng lên toàn bộ trang web.

Tạo file header.php trong folder view/fontend để làm thanh menu

Trang được viết bằng html tuy nhiên lại không có các thẻ <html>,<head>,<body>,… như một trang html thông thường. Bởi vì khi được tải vào trang index.php, toàn bộ nội dung của header.php sẽ nằm trong phần <body> của index.php, vì vậy nó sẽ được áp dụng toàn bộ các tính năng css, jQuery đã được khai báo trong index.php.

Chúng ta sử dụng class navbar có sẵn trong Bootstrap4 để xây dựng menu, các thành phần trong menu sẽ được liệt kê bằng thẻ <ul>, mỗi đối tượng được bao bằng thẻ <li> và sử dụng class nav-item.

Tuy nhiên, do trang web chúng ta xây dựng làm việc chủ yếu trên điện thoại di động, nên việc để nội dung của thanh menu quá dài sẽ gây khó khăn khi sử dụng, mất thẩm mỹ. Ta sẽ dùng class collapse và navbar-collapse để không hiển thị nội dung trong menu khi màn hình ở kích cỡ nhỏ, thay vào đó sẽ hiển thị một nút dạng navbar-toggler-icon (hình 3 dấu gạch ngang) để làm điều khiển menu. Khi nhấn vào nút này, thanh menu sẽ được trình bày như một combobox để người dùng chọn.

Tạo file footer.php trong folder view/fontend: thông tin bản quyền, liên hệ

Tạo file content.php trong folder view/fontend: nội dung trang chủ

Ở đây, chúng ta sử dụng class card trong Bootstrap4. Nội dung của các đối tượng sẽ được gói gọn trong các hình chữ nhật giống các thẻ bài. Nội dung thẻ nằm trong phần card-body, trong đó tiêu đề dùng class card-title sẽ được phóng to, in đậm còn phần nội dung sẽ dùng class card-text. Để có thể nhấp vào thẻ là có thể chuyển đến nội dung, trang web khác luôn mà không cần phải nhấp chính xác vào đường link, ta dùng class streched-link.

Lúc này, các thẻ hiển thị theo chiều ngang gây khó khăn trong việc sử dụng. Ta sẽ sử dụng class card-columns, thì các thẻ sẽ được hiển thị thành 3 cột, mỗi khi thêm một thẻ mới thì các thẻ vẫn tự động sắp xếp. Khi hiển thị trên điện thoại di động, với màn hình nhỏ thì các thẻ sẽ hiển thị theo chiều dọc.

b, Controller

Ta cần cấu hình để khi có người sử dụng thì trang web tự động chuyển đến trang index.php đã tạo.

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

$route['default\_controller'] = 'welcome';

$route['404\_override'] = '';

$route['translate\_uri\_dashes'] = FALSE;

Cấu hình file routes.php trong thư mục config: mặc định khi truy cập sẽ gọi đến controller welcome;

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Welcome extends CI\_Controller

{

public function index() {

$this->load->view('index.php');

}

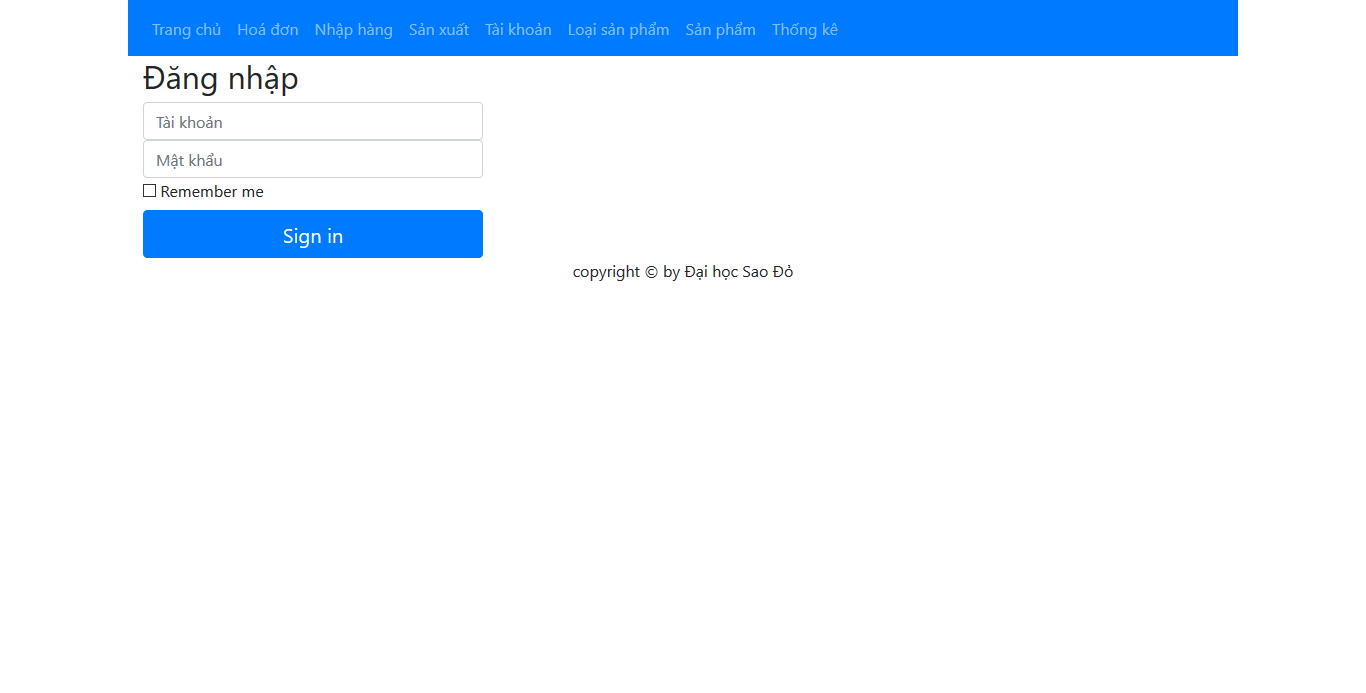
}

Xây dựng controller Welcome.php trong thư mục controller.

Đến đây ta đã hoàn thành phần trang chủ, khi người dùng truy cập vào trang web, controller welcome sẽ được gọi, và controller này sẽ hiển thị view index.php cho người dùng.

### 3.2.2 Chức năng đăng nhập

a, View



Hình . Trang đăng nhập

Tạo file login\_view.php tại folder view

Đầu tiên, chúng ta cần kiểm tra xem người dùng đã đăng nhập hay chưa, nếu có thì chuyển về trang chủ, nếu không thì tiếp tục yêu cầu người dùng nhập vào tài khoản và mật khẩu để đăng nhập.

Tạo form đăng nhập trong thẻ <body>

b, Controller

Tạo file controller Login.php trong thư mục controllers:

<?php

defined('BASEPATH') or exit('No direct script access allowed');

// session\_start();

class Login extends CI\_Controller

{

public function \_\_construct() {

parent::\_\_construct();

// Load database

$this->load->model('login\_database');

}

// Show login page

public function index() {

$this->\_data['mess'] = $this->session->flashdata('flash\_mess');

$this->load->view('login\_view');

}

}

Controller Login có hàm khởi tạo yêu cầu load model login\_database để tương tác với bảng user trong cơ sở dữ liệu. Hàm index sẽ hiển thị cho người dùng trang login\_view.php khi người dùng truy cập vào <http://localhost/login>.

Xây dựng hàm xử lý đăng nhập trong Login controller

// Check for user login process

public function user\_login\_process() {

// Nội dung hàm

}

Cài đặt xác thực form: sau khi người dùng nhấn nút đăng nhập, kiểm tra xem đã nhập tài khoản và mật khẩu chưa. Nếu chưa nhập thì kiểm tra session xem đã đăng nhập trước hay không, nếu có thì trở về trang chủ, nếu không thì hiển thị lại trang đăng nhập.

Nếu đã có thông tin điền vào cả ô tài khoản và mật khẩu thì đóng gói thông tin vào mảng $data để xử lý bằng model. Lưu ý, để đảm bảo tính bảo mật, mật khẩu được mã hoá md5 trước khi gửi đi.

Dùng biến $result để lấy kết quả xử lý từ model, nếu $result là true thì tiếp tục dùng hàm read\_user\_information() trong Login\_Model để lấy các thông tin về tài khoản như tên đăng nhập, loại tài khoản. Không lấy mật khẩu để bảo mật tài khoản. Sau đó lưu thông tin tài khoản đó vào session để không phải đăng nhập mỗi lần sử dụng chức năng. Mặc định session sẽ hết hạn trong 2 tiếng và tự động làm mới sau 5 phút. Nếu $result là false thì báo sai thông tin đăng nhập và chuyển về trang đăng nhập.

Nếu đăng nhập thành công sẽ chuyển về trang chủ, nếu thất bại sẽ trở về trang đăng nhập để người dùng nhập lại thông tin.

c, Model

Tạo model Login\_Database.php trong thư mục models

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Login\_Database extends CI\_Model

{

public function \_\_construct() {

parent::\_\_construct();

}

// Read data using username and password

public function login($data) { }

// Read data from database to show data in admin page

public function read\_user\_information($username) { }

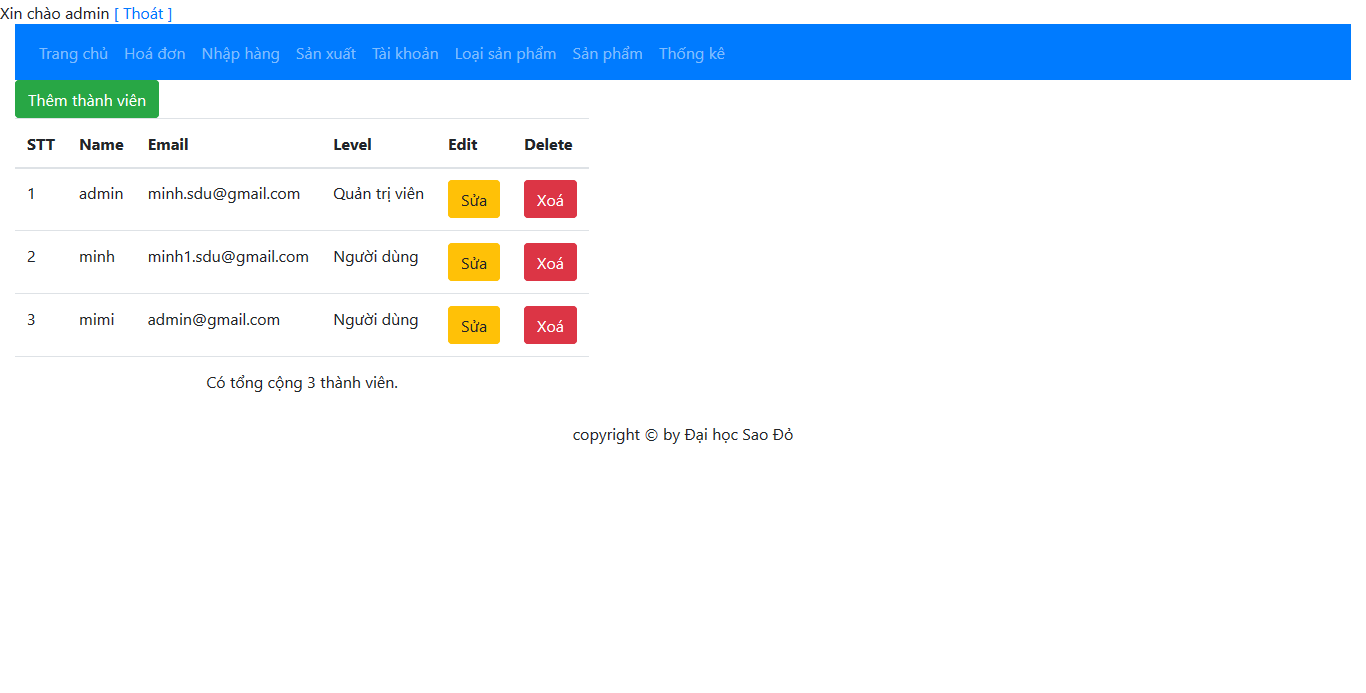
}

Hàm login($data) tìm kiếm tên tài khoản và mật khẩu trong cơ sở dữ liễu, nếu kết quả trả về chính xác bằng 1 thì trả về true, nếu không thì trả về false.

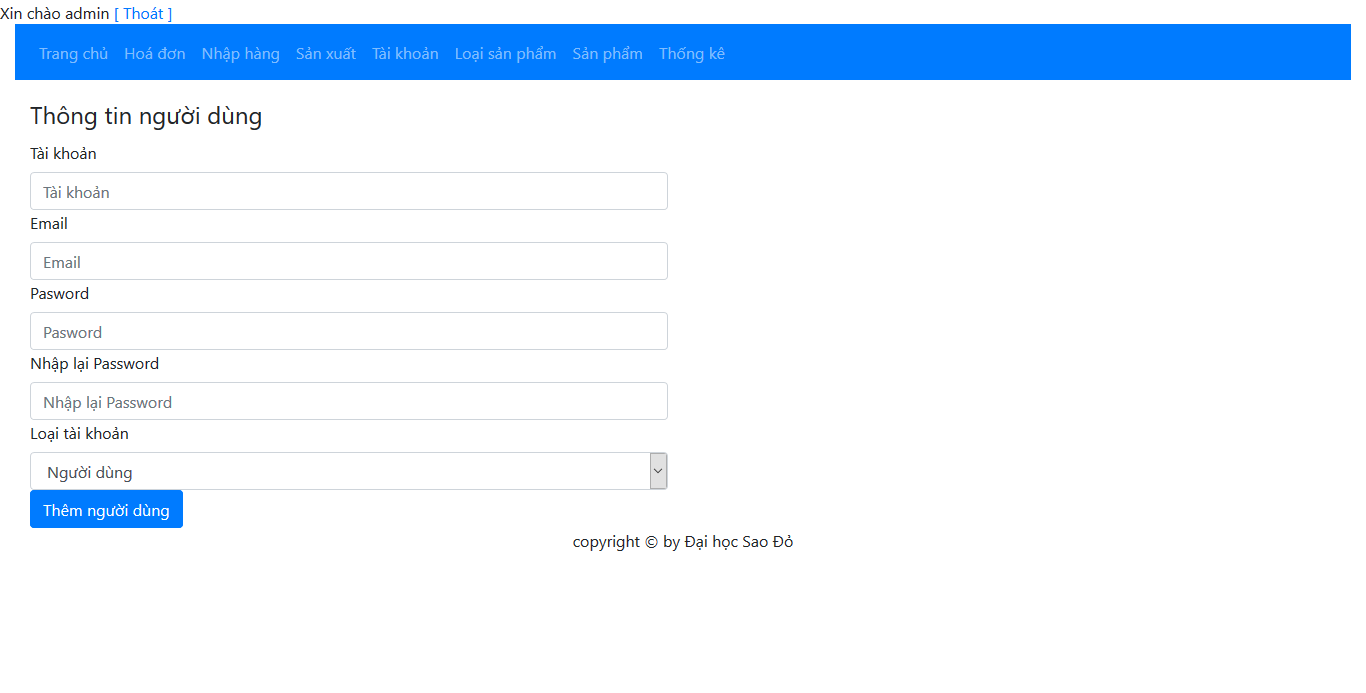
Hàm read\_user\_information($username) trả về thông tin tài khoản lưu trong cơ sở dữ liệu.

### 3.2.3 Chức năng quản lý tài khoản

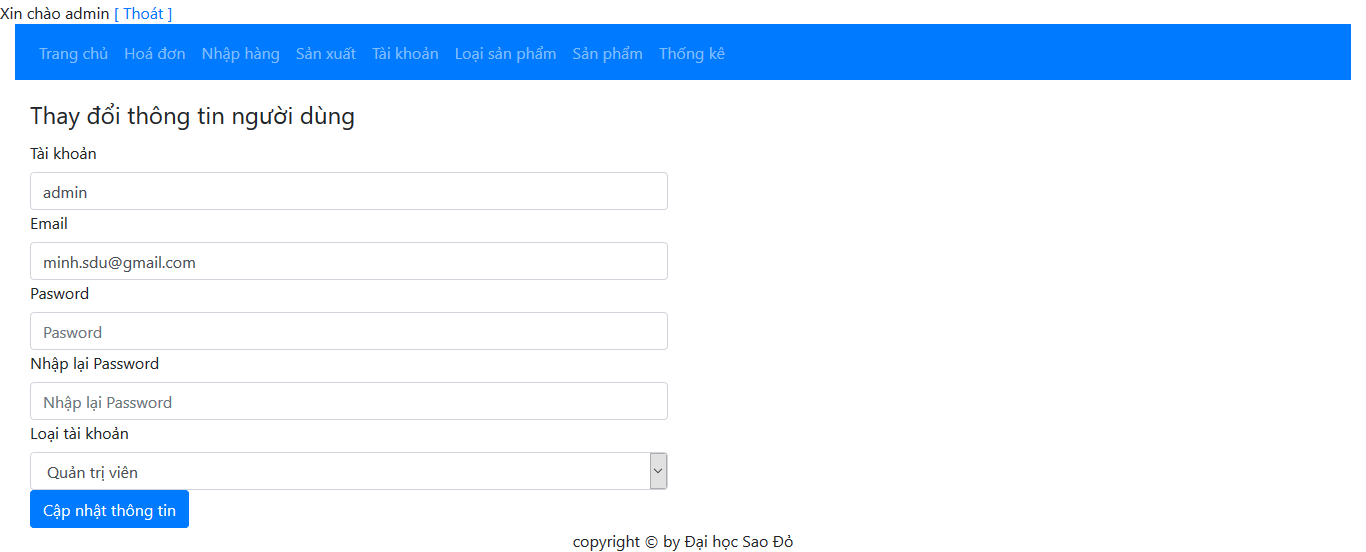
a, View



Hình . Trang thông tin tài khoản



Hình . Trang thêm tài khoản



Hình . Trang sửa thông tin tài khoản

Để truy cập vào trang quản lý tài khoản người dùng cần đăng nhập tài khoản của mình. Chúng ta sẽ tạo một file status.php trong folder fontend để có thể tái sử dụng lại trong các chức năng.

<?php

if (isset($this->session->userdata['logged\_in'])) {

$username = ($this->session->userdata['logged\_in']['username']);

echo "Xin chào " . $username . " ";

echo "<a href= " . base\_url() . "index.php/login/logout>[ Thoát ]</a>";

} else {

header("location: login");

}

?>

File này sẽ kiểm tra xem đã có tài khoản đăng nhập trong session chưa, nếu có thì hiển thị câu chào thông báo, nếu không sẽ chuyển hướng sang trang đăng nhập.

Trong folder views, tạo folder user để chứa những file view cho chức năng quản lý tài khoản: main.php, index\_view.php, add\_view.php, edit\_view.php

Tạo file main.php: đây là file mà User controller sẽ trỏ đến mỗi khi có hàm được thực hiện. Nội dung cần hiển thị sẽ được nhúng vào thông qua lệnh:

<?php $this->load->view($subview); ?>

Đầu tiên cần kiểm tra thông tin đăng nhập bằng file status.php vừa tạo.

<?php $this->load->view("fontend/status"); ?>

Tiêu đề trang sẽ được thêm vào từ controller:

<title><?php echo $title; ?></title>

<div class="container-fluid">

<?php $this->load->view("fontend/header"); ?>

<div id="content">

<?php $this->load->view($subview); ?>

</div>

<?php $this->load->view("fontend/footer"); ?>

</div>

Tải giao diện trang từ folder fontend, và nội dung trang

Trang main.php hoàn chỉnh như sau:

Tạo trang index\_view.php: liệt kê danh sách các tài khoản, nơi chọn chức năng thêm, sửa, xoá tài khoản

Tạo trang add\_view.php trong folder user để làm nội dung cho chức năng thêm tài khoản

Tạo trang edit\_view.php trong folder user để làm nội dung cho chức năng chỉnh sửa thông tin tài khoản

b, Controller

Tạo file controller User.php trong folder controllers để điều khiển chức năng quản lý tài khoản:

Tại hàm khởi tạo, chúng ta load model User\_Model để tương tác với cơ sở dữ liệu. Hàm index chạy khi người dùng truy cập vào chức năng quản lý tài khoản. Hàm này trỏ về file main.php. Trong đó đã được đặt tiêu đề trang, trang nội dung, danh sách tài khoản được lấy từ cơ sở dữ liệu và thông báo nếu có.

Thêm hàm xoá tài khoản vào controller

public function del($id) {

$this->User\_Model->deleteUser($id);

$this->session->set\_flashdata("flash\_mess", "Delete Success");

redirect(base\_url() . "user");

}

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class User extends CI\_Controller

{

protected $data;

public function \_\_construct() {

parent::\_\_construct();

$this->load->helper('url');

$this->load->model('User\_Model');

}

public function index() { }

}

Hàm này sẽ xoá tài khoản dựa theo id, sau đó trở về trang quản lý tài khoản với tin nhắn đã xoá thành công.

Thêm hàm kiểm tra tên đăng nhập để kiểm soát trùng tên đăng nhập khi thêm hoặc sửa tài khoản

public function check\_user($user, $id) { }

public function add() { }

Thêm hàm chèn tài khoản mới. Đầu tiên, thêm tiêu đề và nội dung cho trang thêm tài khoản. Sau đó kiểm tra các ô

Thêm hàm kiểm tra email để kiểm soát trùng email

Thêm hàm chỉnh sửa tài khoản

public function edit($id) { }

public function check\_email($email, $id) { }

c, Model

Tạo file User\_Model.php trong thư mục models để làm việc với bảng user trong cơ sở dữ liệu

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class User\_Model extends CI\_Model

{

protected $table = 'user';

public function \_\_construct() {

parent::\_\_construct();

//Do your magic here

}

public function getList(){ }

public function countAll(){ }

public function deleteUser($id){ }

public function insertUser($data\_insert){ }

public function getUserById($id){ }

}

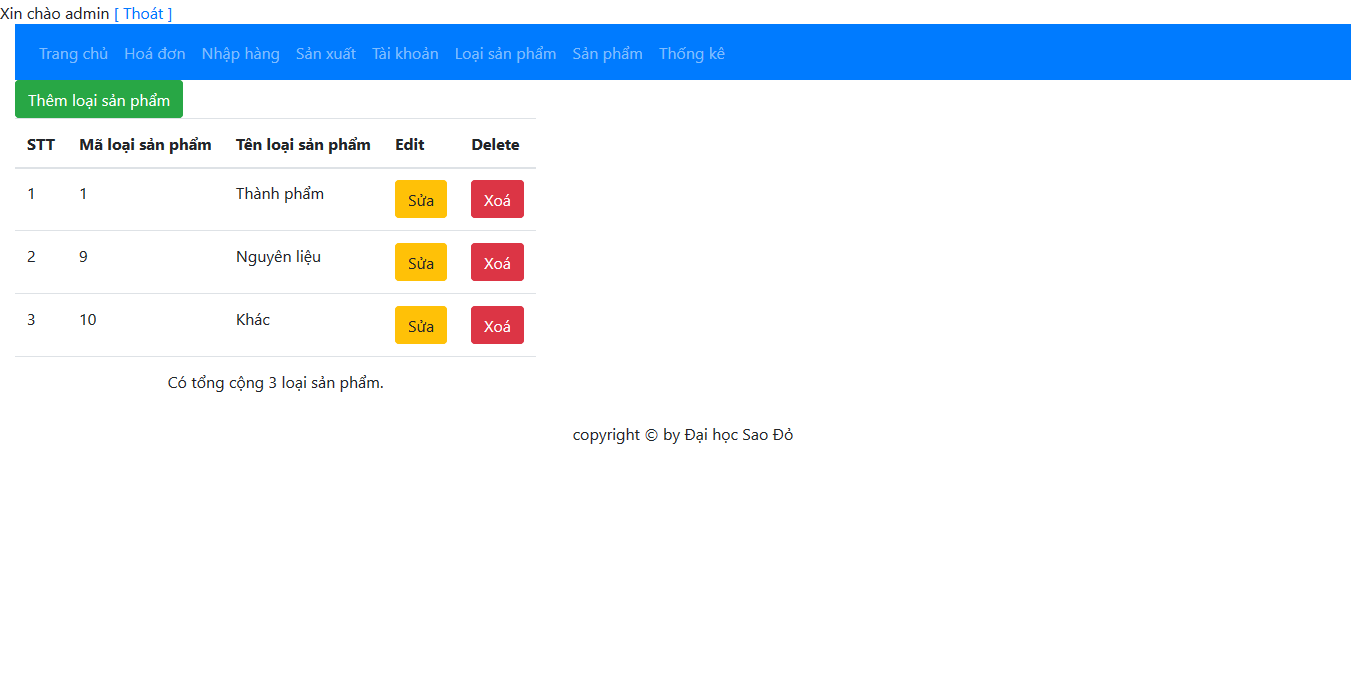
public function updateUser($data\_update, $id) { }

public function checkUsername($user, $id = "") { }

public function checkEmail($email, $id = "") { }

### 3.2.4 Chức năng quản lý loại sản phẩm

A, View



Hình . Trang quản lý loại sản phẩm

Trong folder views, tạo folder loaisanpham để chứa những file view cho chức năng quản lý loại sản phẩm: main.php, index\_view.php, add\_view.php, edit\_view.php

Tạo file main.php: đây là file mà Loaisanpham controller sẽ trỏ đến mỗi khi có hàm được thực hiện. Nội dung của file này giống như file main.php trong chức năng quản lý tài khoản.

Tạo file index\_view.php hiển thị nội dung khi truy cập vào chức năng quản lý loại sản phẩm.

Tạo file add\_view.php hiển thị nội dung khi chọn chức năng thêm loại sản phẩm

Tạo file edit\_view.php hiển thị nội dung khi chọn chức năng chỉnh sửa loại sản phẩm

b, Controller

Trong folder controllers tạo file Loaisanpham.php để điều khiển chức năng quản lý loại sản phẩm.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Loaisanpham extends CI\_Controller

{

protected $data;

public function \_\_construct() {

parent::\_\_construct();

$this->load->model('Loaisanpham\_Model');

//Do your magic here

}

public function index() {

$this->\_data['subview'] = 'loaisanpham/index\_view.php';

$this->\_data['title'] = 'Danh sách loại sản phẩm';

$this->\_data['info'] = $this->Loaisanpham\_Model->getList();

$this->\_data['total\_loaisanpham'] = $this->Loaisanpham\_Model->countAll();

$this->\_data['mess'] = $this->session->flashdata('flash\_mess');

$this->load->view('loaisanpham/main.php', $this->\_data);

}

}

Thêm hàm xoá loại sản phẩm:

public function del($id) {

$this->Loaisanpham\_Model->deleteLoaisanpham($id);

$this->session->set\_flashdata("flash\_mess", "Delete Success");

redirect(base\_url() . "loaisanpham");

}

Thêm hàm chèn loại sản phẩm mới:

public function add() { }

Thêm hàm chỉnh sửa loại sản phẩm

public function edit($id) { }

c, Model

Trong thư mục models, tạo file Loaisanpham\_Model.php để làm việc với bảng loaisanpham trong cơ sở dữ liệu:

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Loaisanpham\_Model extends CI\_Model

{

protected $table = 'loaisanpham';

public function \_\_construct() {

parent::\_\_construct();

}

public function getList() { }

public function countAll() { }

public function deleteLoaisanpham($id) { }

public function insertLoaisanpham($data\_insert) { }

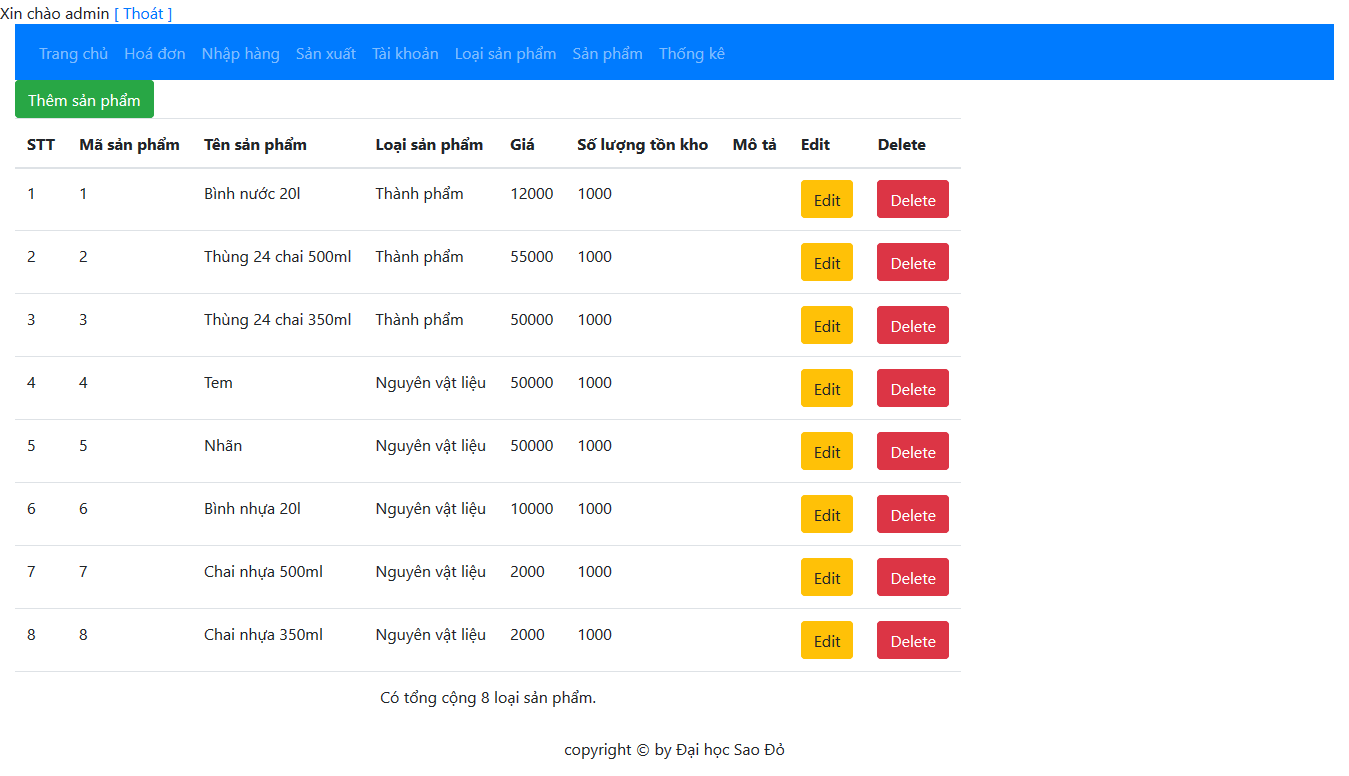
public function getLoaisanphamById($id) { }

public function updateLoaisanpham($data\_update, $id) { }

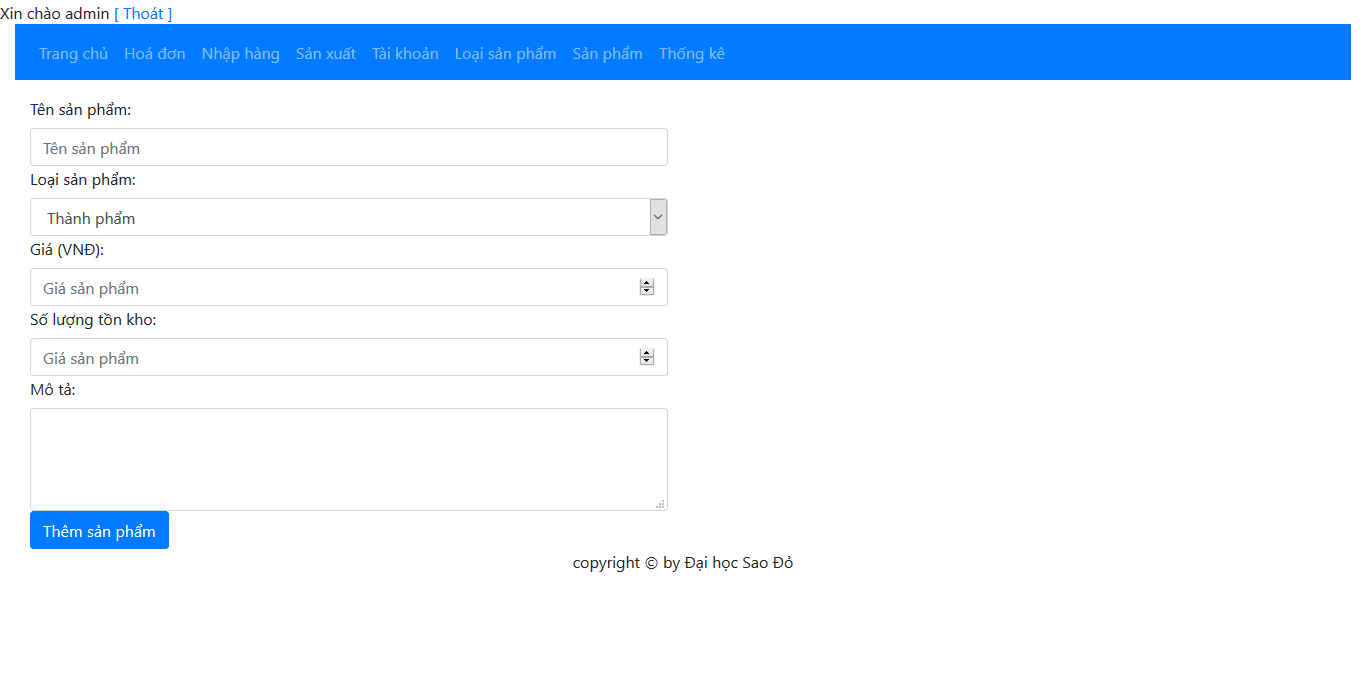
}

### 3.2.5 Chức năng quản lý sản phẩm

A, View



Hình . Trang quản lý sản phẩm



Hình . Trang thêm sản phẩm mới

Trong folder views, tạo folder sanpham để lưu file hiển thị nội dung cho chức năng quản lý sản phẩm. Tạo lần lượt các file main.php, index\_view.php, add\_view.php, edit\_view.php trong folder sanpham.

File main.php có nội dung tương tự file main.php trong chức năng quản lý tài khoản.

File index\_view.php: hiển thị nội dung khi truy cập vào trang quản lý tài khoản

File add\_view.php: nội dung trang thêm sản phẩm mới

File edit\_view.php: nội dung trang chỉnh sửa thông tin sản phẩm

b, Controller

Tạo file Sanpham.php trong folder controllers để điều khiển chức năng quản lý sản phẩm.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Sanpham extends CI\_Controller

{

protected $\_data;

public function \_\_construct() {

parent::\_\_construct();

$this->load->model('SanPham\_Model', 'SP\_Model');

}

public function index() { }

public function del($id) { }

}

Thêm hàm chèn sản phẩm mới

public function add() { }

Thêm hàm chỉnh sửa thông tin sản phẩm

public function edit($id) { }

c, Model

Tạo file Sanpham\_Model.php trong thư mục models để làm việc với bảng sanpham trong cơ sở dữ liệu

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class SanPham\_Model extends CI\_Model

{

protected $table = 'sanpham';

public function \_\_construct(){

parent::\_\_construct();

}

public function getList() { }

public function getSanPhamById($id) { }

public function countAll() { }

public function deleteSanpham($id) { }

public function insertSanPham($data\_insert) { }

public function updateSanPham($data\_update, $id) { }

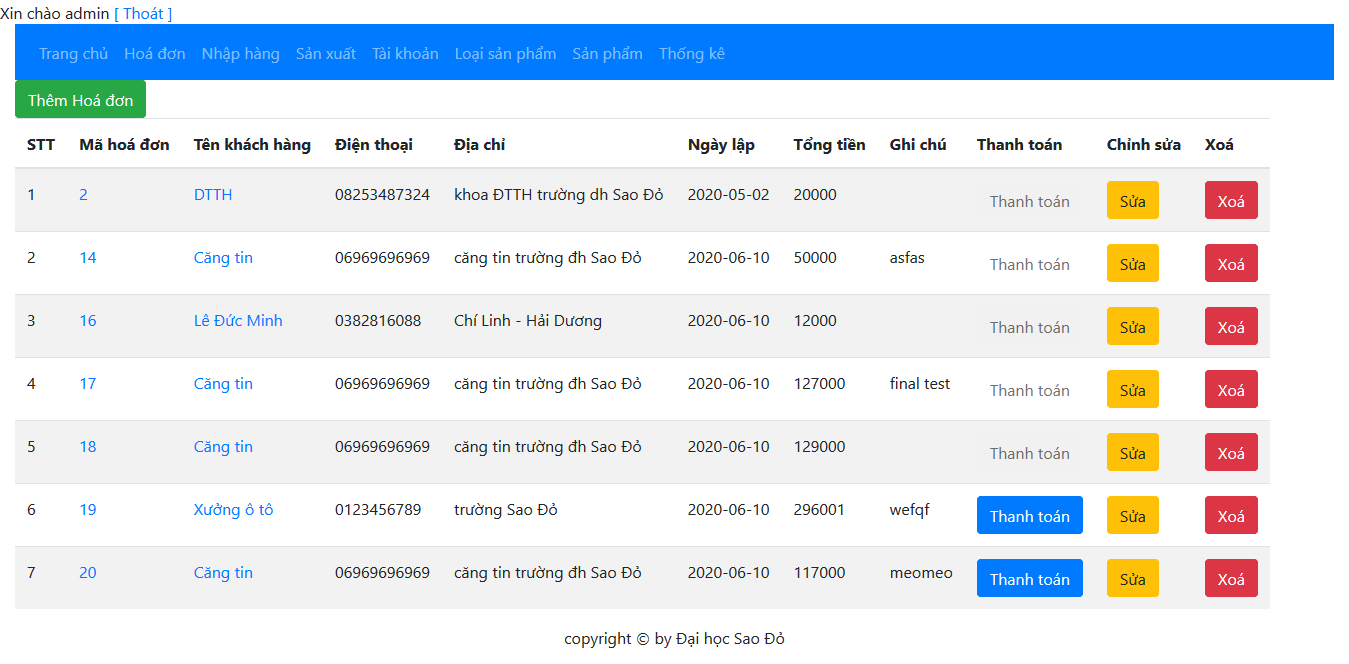
public function getThanhPham() { }

public function getNguyenLieu() { }

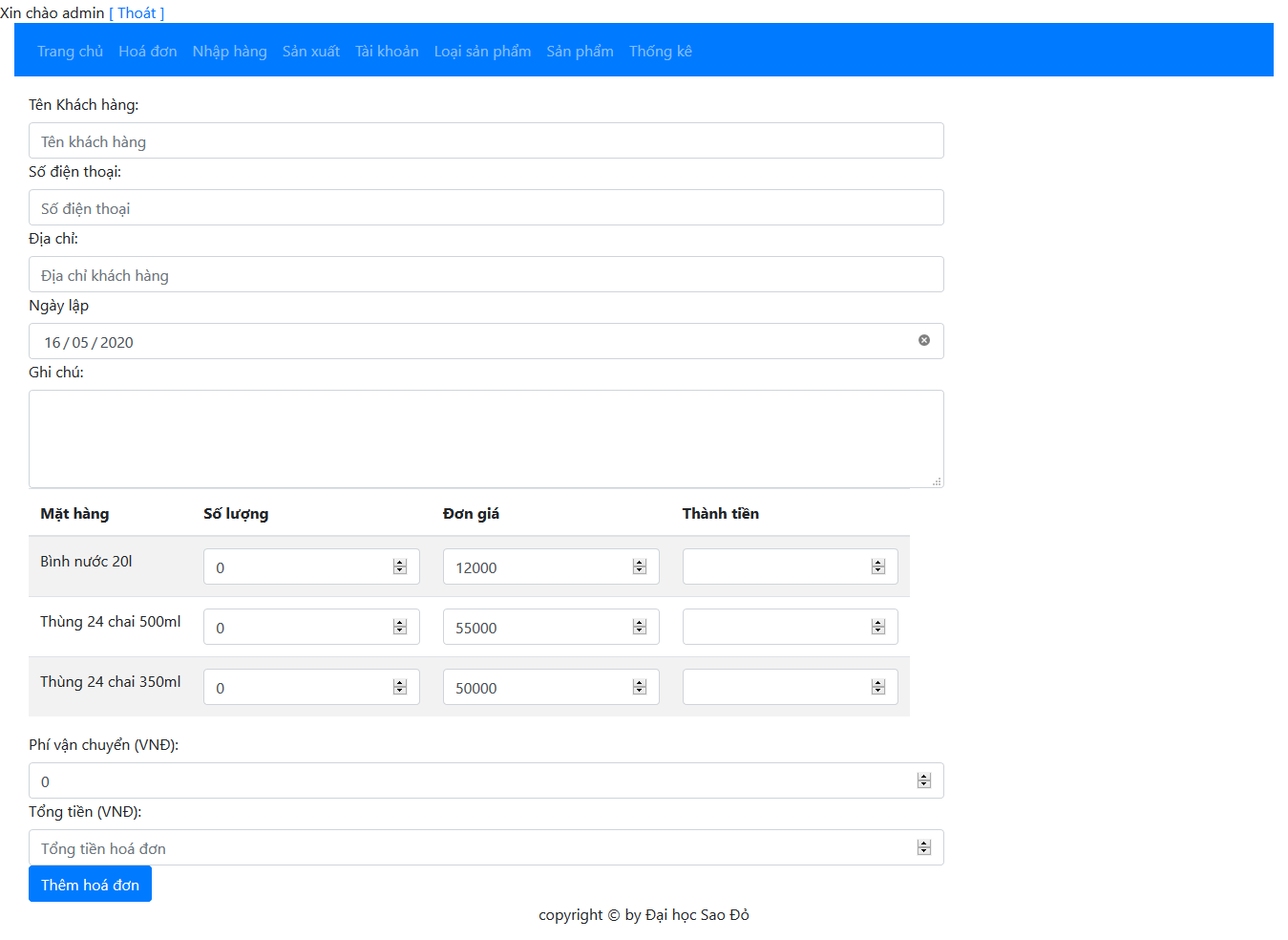
}

### 3.2.6 Chức năng quản lý hoá đơn

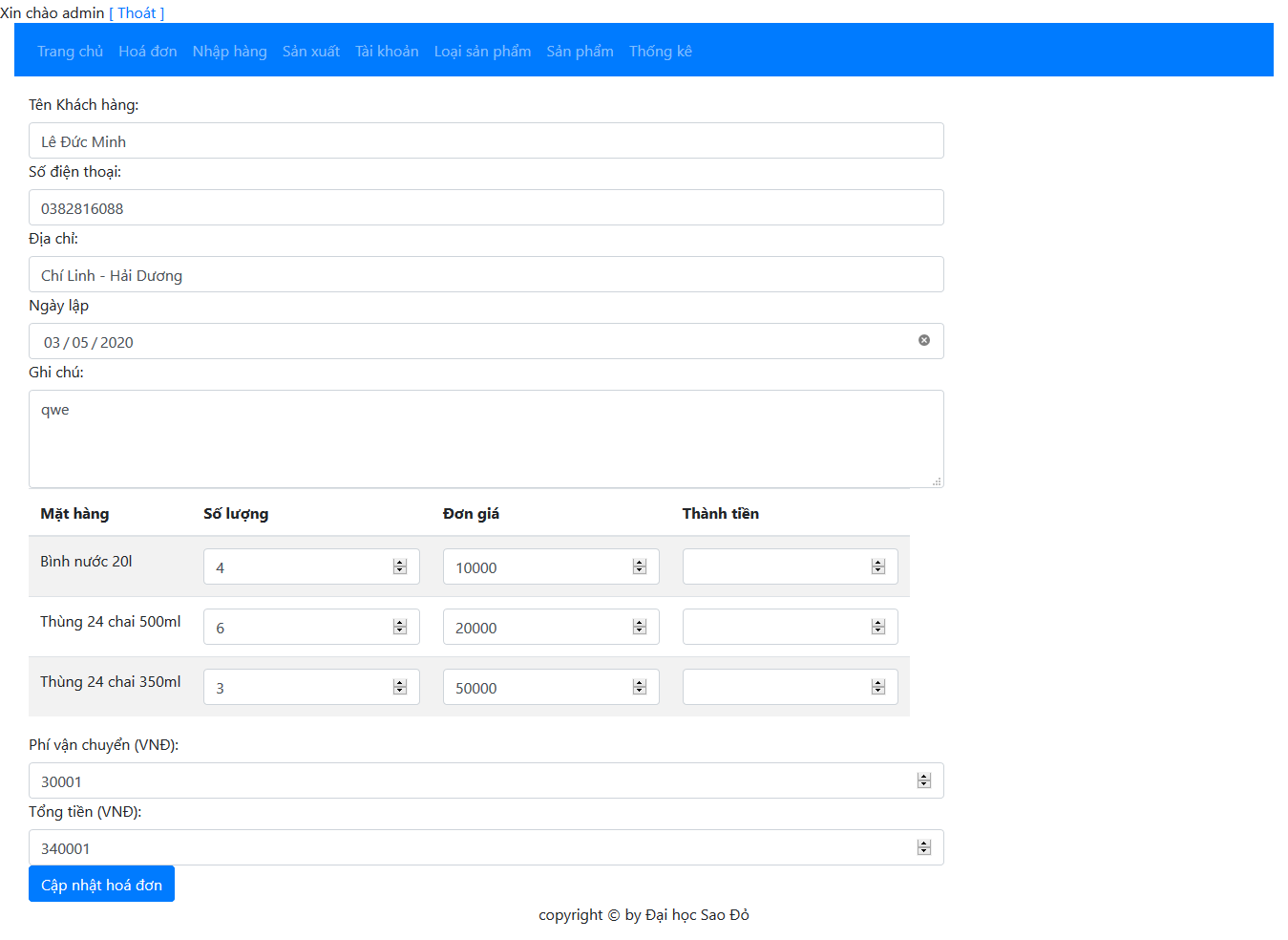
a, View



Hình . Trang quản lý hoá đơn



Hình . Trang thêm hoá đơn mới



Hình . Trang chỉnh sửa hoá đơn

Trong folder views tạo folder hoadon để lưu các file nội dung hiển thị chức năng quản lý hoá đơn. Tiếp tục tạo các file main.php, index\_view.php, add\_view.php, edit\_view.php trong folder hoadon. File main.php có nội dung tương tự file main.php trong chức năng quản lý tài khoản.

Scrip tính tiền tự động:

<!-- scrip tính tiền tự động -->

<script>

function tinh(id1, id2, id3) {

var x = document.getElementById(id1).value \* document.getElementById(id2).value;

document.getElementById(id3).value = x;

tien();

}

function tien() {

var i = 0;

var tt = 0;

<?php foreach ($info as $item) { ?>

if (document.getElementById("soluong<?php echo $item['id']; ?>").value != 0) {

tt = parseInt(tt) + parseInt(document.getElementById("soluong<?php echo $item['id'] ?>").value \* document.getElementById("gia<?php echo $item['id'] ?>").value);

}

<?php } ?>

tt = parseInt(tt) + parseInt(document.getElementById("phivanchuyen").value);

document.getElementById("tongtien").value = tt;

}

</script>

File edit\_view.php cũng yêu cầu chèn đoạn script tính tiền tự động như file add\_view.php

b, Controller

Tạo file Hoadon.php trong folder controllers để điều khiển chức năng quản lý hoá đơn

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Hoadon extends CI\_Controller

{

protected $\_data;

public function \_\_construct() {

parent::\_\_construct();

//Do your magic here

$this->load->model('Hoadon\_Model', 'HD\_Model');

$this->load->model('ChitietHoadon\_Model', 'CTHD\_Model');

$this->load->model('SanPham\_Model', 'SP\_Model');

}

public function index() {

$this->\_data['subview'] = 'hoadon/index\_view.php';

$this->\_data['title'] = 'Danh sách Hoá đơn';

$this->\_data['info'] = $this->HD\_Model->getList();

$this->\_data['total\_hoadon'] = $this->HD\_Model->countAll();

$this->\_data['mess'] = $this->session->flashdata('flash\_mess');

$this->load->view('hoadon/main.php', $this->\_data);

}

public function del($id) {

$this->HD\_Model->deleteHoadon($id);

$this->CTHD\_Model->deleteCTHDByMahoadon($id);

$this->session->set\_flashdata("flash\_mess", "Delete Success");

redirect(base\_url() . "index.php/hoadon");

}

}

Thêm hàm xử lý chèn hoá đơn:

public function add() { }

Thêm hàm xử lý sửa hoá đơn:

public function edit($id) { }

c, Model

Do chức năng hoá đơn cần sử dụng 2 bảng: hoadon và chitiethoadon nên chúng ta cần tạo 2 model trong folder models để làm việc với 2 bảng trong cơ sở dữ liệu:

File Hoadon\_Model.php

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Hoadon\_Model extends CI\_Model

{

protected $\_table = 'hoadon';

public function \_\_construct(){

parent::\_\_construct();

}

public function getList(){ }

public function getHoadonById($id){ }

public function getHoadonMoinhat() { }

public function countAll() { }

public function deleteHoadon($id) { }

public function insertHoadon($data\_insert) { }

public function updateHoadon($data\_update, $id) { }

}

File ChitietHoadon\_Model.php:

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ChitietHoadon\_Model extends CI\_Model

{

protected $\_table = 'chitiethoadon';

public function \_\_construct() {

parent::\_\_construct();

}

public function getList() { }

public function getCTHDById($id) { }

public function countAll() { }

public function deleteCTHD($id) { }

public function insertCTHD($data\_insert) { }

public function updateCTHD($data\_update, $id) { }

public function getCTHDByMahoadon($id) { }

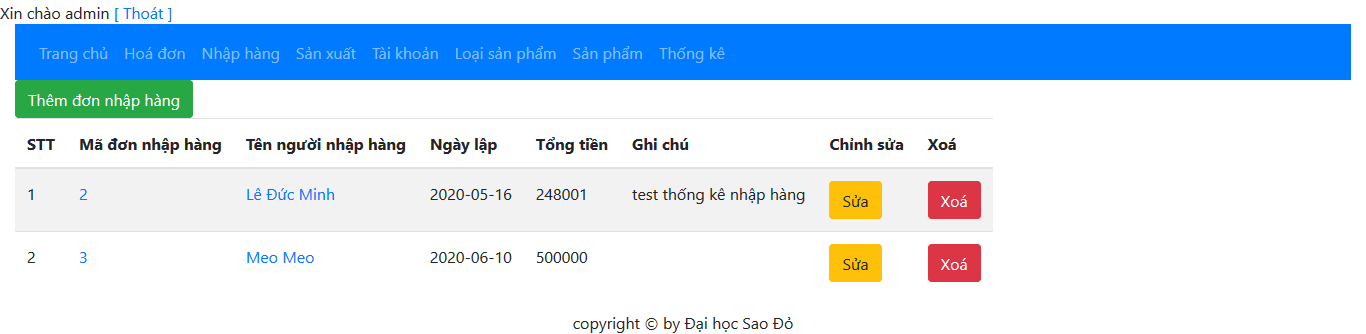
public function getCTHDForUpdate($mahoadon, $masanpham) { }

public function deleteCTHDByMahoadon($mahoadon) { }

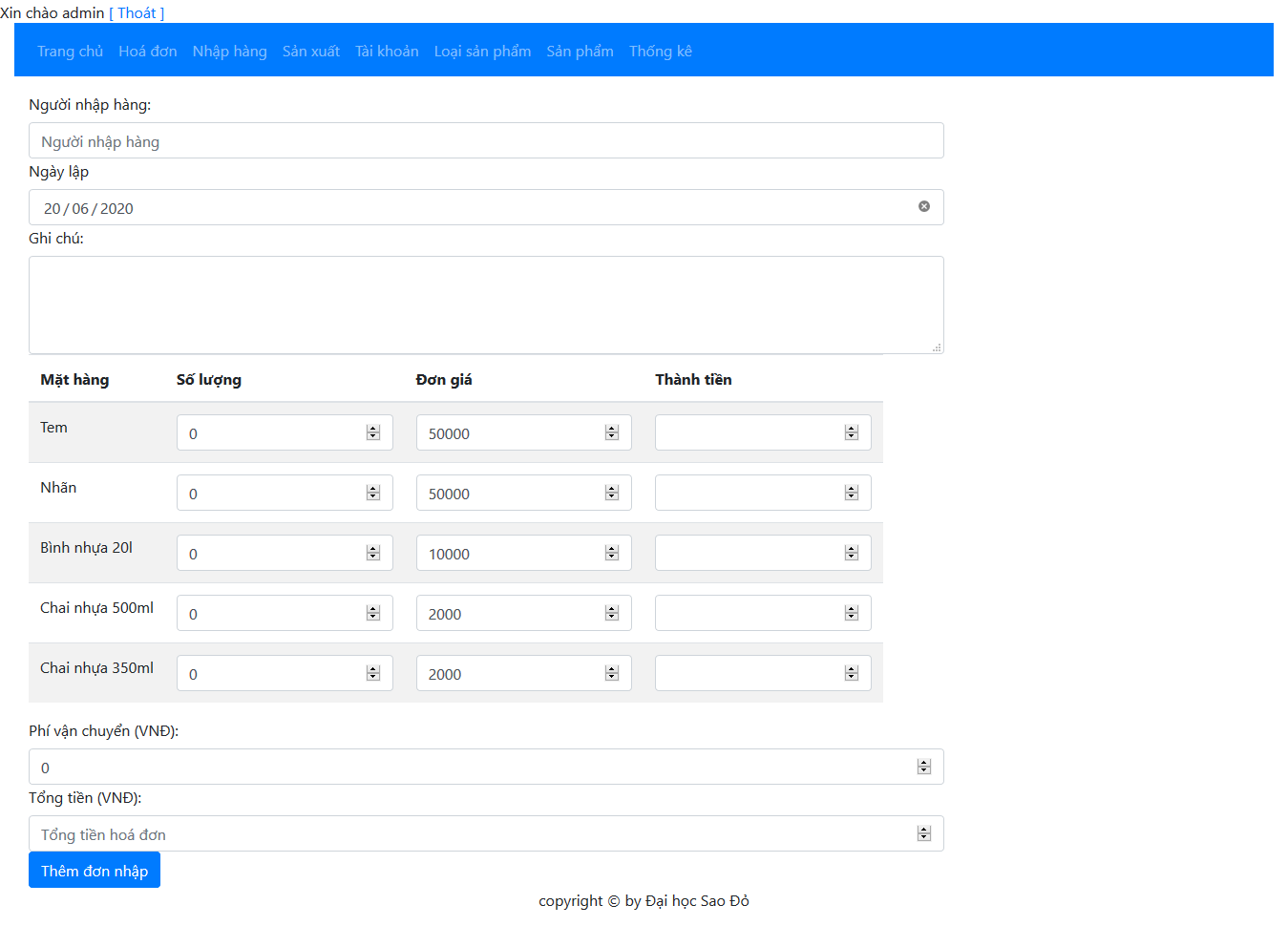
}

### 3.2.7 Chức năng quản lý nhập hàng

a, View



Hình . Trang quản lý nhập hàng



Hình . Trang thêm đơn nhập hàng

Trong folde views tạo folder nhaphang chứa các file hiển thị nội dung chức năng quản lý nhập hàng: main.php, index\_view.php, add\_view.php, edit\_view.php. File main.php có nội dung tương tự như file main.php của chức năng quản lý tài khoản.

File add\_view.php cần chèn thêm đoạn scripts tính tiền tự động trong chức năng quản lý hoá đơn

File edit\_view.php cần chèn thêm đoạn scripts tính tiền tự động trong chức năng quản lý hoá đơn.

b, Controller

Trong folder controllers tạo file Nhaphang.php để điều khiển chức năng quản lý nhập hàng.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Nhaphang extends CI\_Controller

{

protected $\_data;

public function \_\_construct() {

parent::\_\_construct();

$this->load->model('Nhaphang\_Model', 'NH\_Model');

$this->load->model('ChitietNhaphang\_Model', 'CTNH\_Model');

$this->load->model('SanPham\_Model', 'SP\_Model');

}

public function index() {

$this->\_data['subview'] = 'nhaphang/index\_view.php';

$this->\_data['title'] = 'Danh sách Nhập hàng';

$this->\_data['info'] = $this->NH\_Model->getList();

$this->\_data['total\_nhaphang'] = $this->NH\_Model->countAll();

$this->\_data['mess'] = $this->session->flashdata('flash\_mess');

$this->load->view('nhaphang/main.php', $this->\_data);

}

public function del($id) {

$this->NH\_Model->deleteNhaphang($id);

$this->CTNH\_Model->deleteCTNHBymanhaphang($id);

$this->session->set\_flashdata("flash\_mess", "Delete Success");

redirect(base\_url() . "nhaphang");

}

public function add() { }

public function edit($id) { }

c, Model

Trong folder models tạo file Nhaphang\_Model.php, và file ChitietNhaphang\_Model.php. Nội dung file Nhaphang\_Model.php

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Nhaphang\_Model extends CI\_Model

{

protected $\_table = 'nhaphang';

public function \_\_construct() {

parent::\_\_construct();

}

public function getList() { }

public function getNhaphangById($id) { }

public function getNhaphangMoinhat() { }

public function countAll() { }

public function deleteNhaphang($id) { }

public function insertNhaphang($data\_insert) { }

public function updateNhaphang($data\_update, $id) { }

}

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ChitietNhaphang\_Model extends CI\_Model

{

protected $\_table = 'chitietnhaphang';

public function \_\_construct(){

parent::\_\_construct();

}

public function getList(){ }

public function getCTNHById($id){ }

public function countAll(){ }

public function deleteCTNH($id){ }

public function insertCTNH($data\_insert){ }

public function updateCTNH($data\_update, $id){ }

public function getCTNHByManhaphang($id){ }

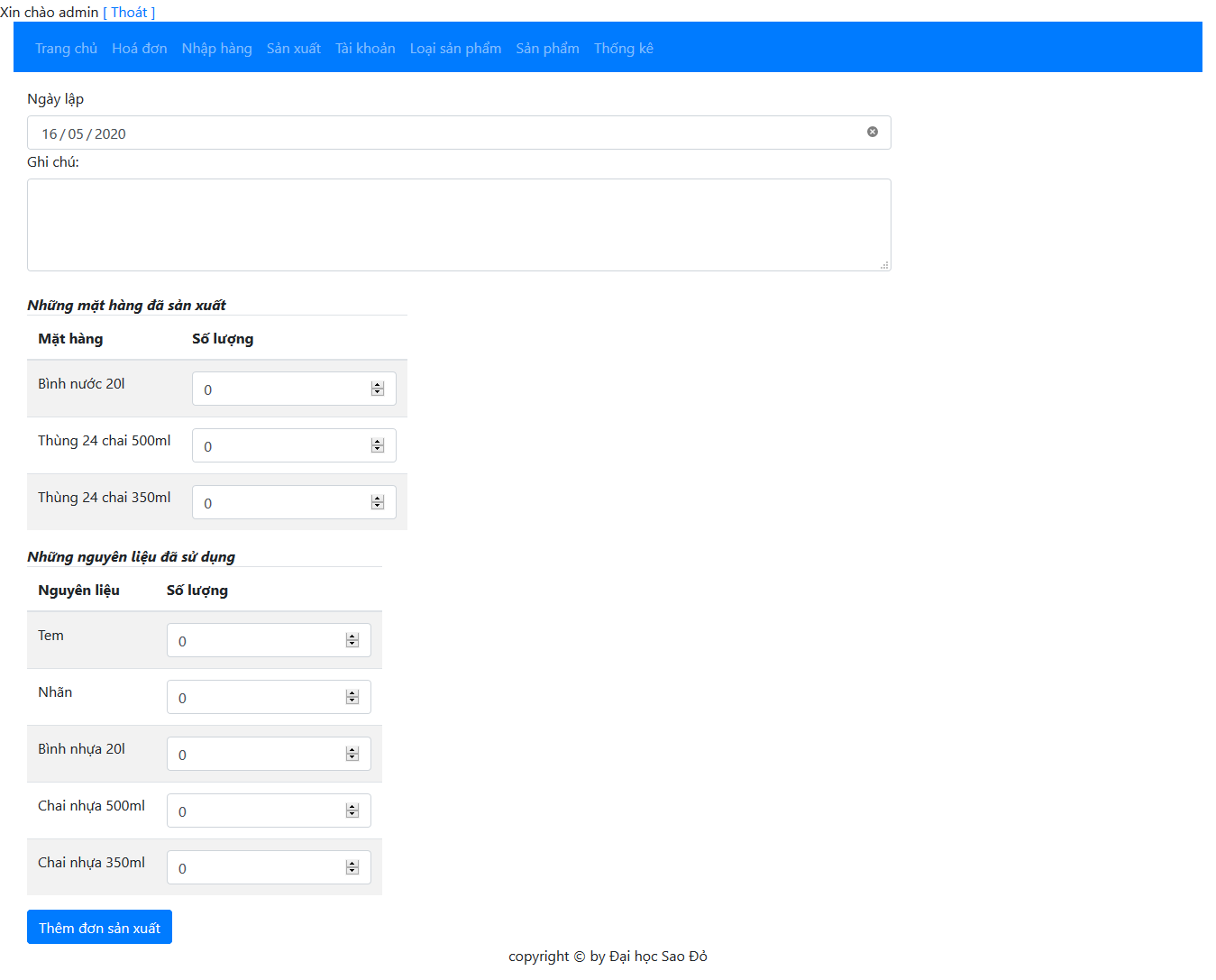
public function getCTNHForUpdate($manhaphang, $masanpham){ }

public function deleteCTNHBymanhaphang($manhaphang) { }

}

### 3.2.8 Chức năng quản lý sản xuất

a, View



Hình . Trang thêm đơn sản xuất

Trong folder views tạo folder sanxuat chưa các file: main.php, index\_view.php, add\_view.php, edit\_view.php. File main.php có nội dung tương tự như file main.php trong chức năng quản lý tài khoản.

File add\_view.php và edit\_view.php cần chèn thêm đoạn scripts tự động tính tiền như chức năng quản lý hoá đơn

b, Controller

Tạo file Sanxuat.php điều khiển chức năng quản lý sản xuât

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Sanxuat extends CI\_Controller

{

protected $\_data;

public function \_\_construct() {

parent::\_\_construct();

$this->load->model('SanPham\_Model', 'SP\_Model');

$this->load->model('Sanxuat\_Model', 'SX\_Model');

$this->load->model('ChitietSanxuat\_Model', 'CTSX\_Model');

}

public function index() {

$this->\_data['subview'] = 'sanxuat/index\_view.php';

$this->\_data['title'] = 'Lịch sử sản xuất';

$this->\_data['info'] = $this->SX\_Model->getList();

$this->\_data['total\_sanxuat'] = $this->SX\_Model->countAll();

$this->\_data['mess'] = $this->session->flashdata('flash\_mess');

$this->load->view('sanxuat/main.php', $this->\_data);

}

public function del($id){

}

public function add(){

}

public function edit($id) {

}

c, Model

Trong folder models tạo file Sanxuat\_Model.php và ChitietSanxuat\_Model.php.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Sanxuat\_Model extends CI\_Model

{

protected $\_table = 'sanxuat';

public function \_\_construct() {

parent::\_\_construct();

}

public function getList() { }

public function getSanxuatById($id) { }

public function getSanxuatMoinhat() { }

public function countAll() { }

public function deleteSanxuat($id) { }

public function insertSanxuat($data\_insert) { }

public function updateSanxuat($data\_update, $id) { }

}

File ChitietSanxuat\_Model.php

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ChitietSanxuat\_Model extends CI\_Model

{

protected $\_table = 'chitietsanxuat';

public function \_\_construct(){

parent::\_\_construct();

}

public function getList(){ }

public function getCTSXById($id){ }

public function countAll(){ }

public function deleteCTSX($id){ }

public function insertCTSX($data\_insert){ }

public function updateCTSX($data\_update, $id){ }

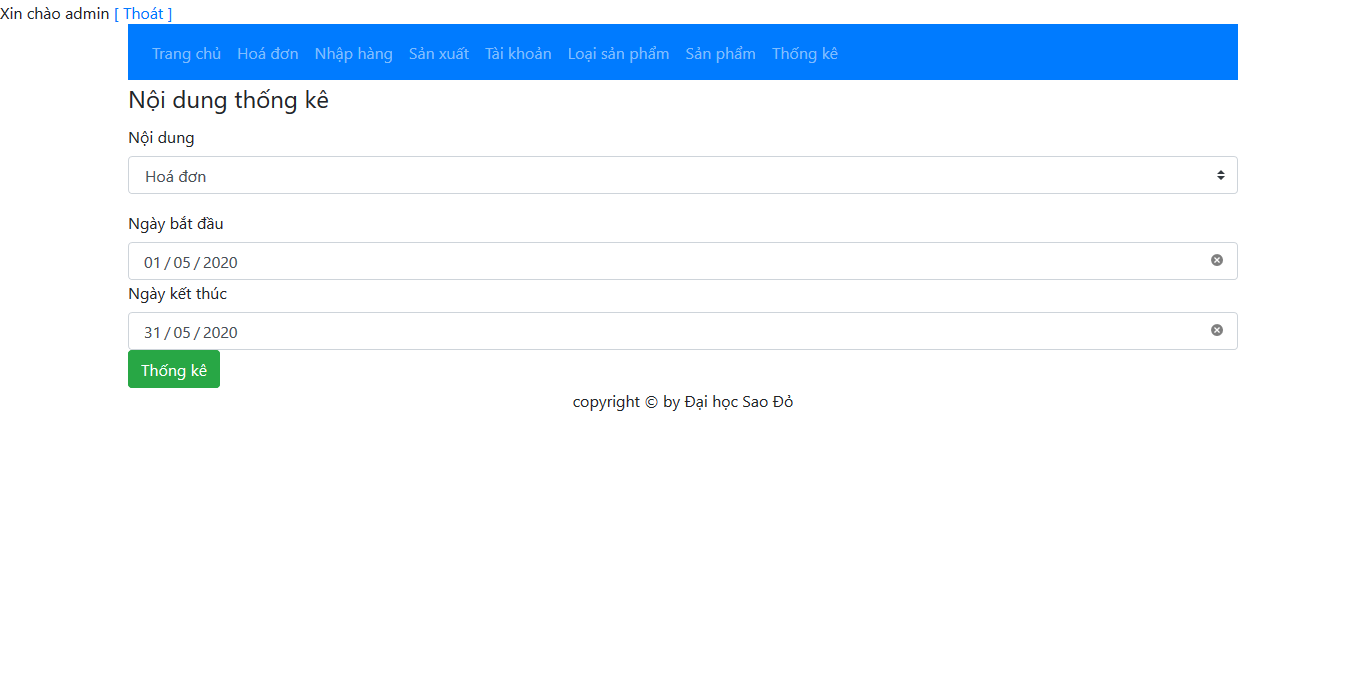
public function getCTSXByMasanxuat($id){ }

public function getCTSXForUpdate($masanxuat, $masanpham){}

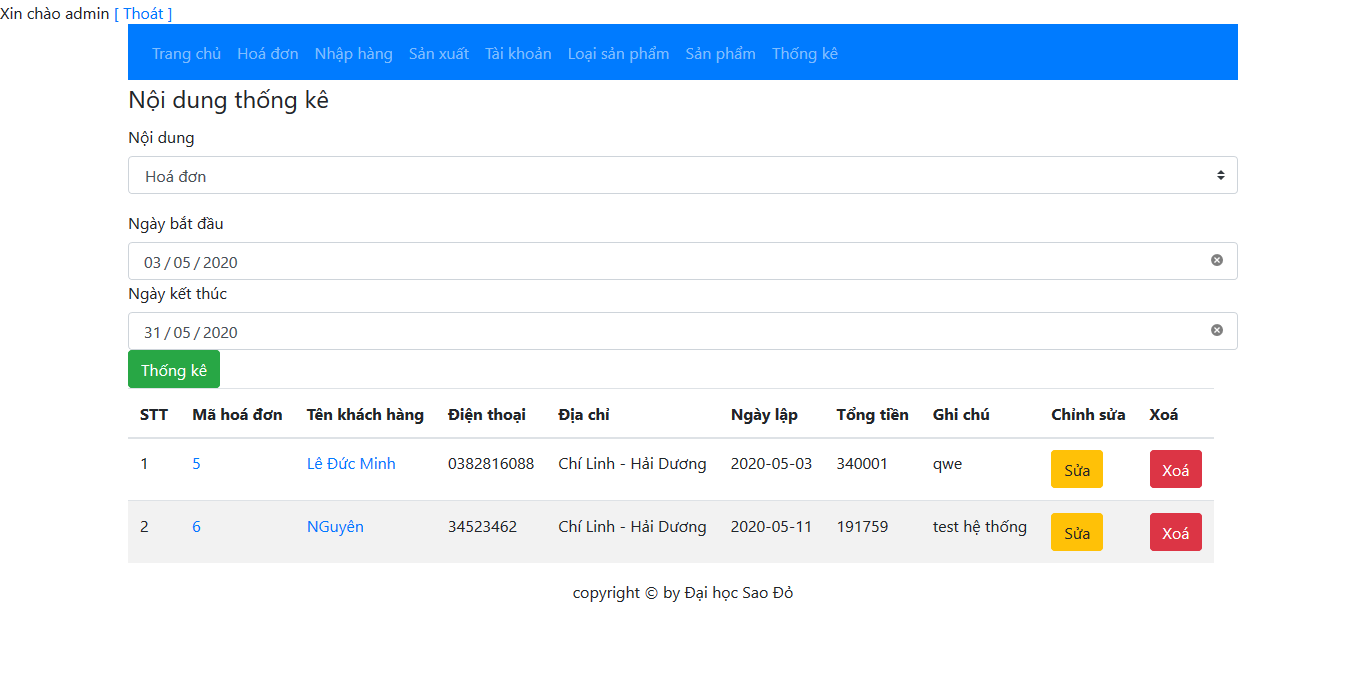
}

### 3.2.9 Chức năng thống kê

a, View



Hình . Trang thống kê



Hình . Ví dụ thống kê hoá đơn

Tạo folder thongke trong thư mục views. Tạo các file sau để hiển thị nội dung cho chức năng thống kê. File main.php tương tự như chức năng quản lý tài khoản.

* File index\_view.php
* File hoadon\_view.php
* File nhaphang\_view.php
* File sanxuat\_view.php

b, Controller

Trong folder controllers tạo file Thongke.php để điều khiển chức năng thống kê.

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class Thongke extends CI\_Controller

{

protected $\_data;

public function \_\_construct() {

parent::\_\_construct();

$this->load->model('Thongke\_Model', 'TK\_Model');

$this->load->model('ChitietHoadon\_Model', 'CTHD\_Model');

}

public function index() {

$this->\_data['title'] = 'Thống kê';

$this->\_data['subview'] = "thongke/index\_view.php";

$this->load->view('thongke/main.php', $this->\_data);

}

public function getview() {

}

}

Do chức năng thống kê sử dụng các bảng đã được tạo model nên ta không cần tạo model riêng cho chức năng này mà chỉ cần load các model được tạo trước đó trong hàm \_\_construct().

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1. Kết luận**

**1.1. Kết quả đạt được**

* Trang web có giao diện thân thiện, dễ sử dụng .
* Trang web có các chức năng phù hợp yêu cầu: quản lý sản phẩm, quản lý hoá đơn,…
* Chức năng nhập và quản lý được thực hiện logic, đảm bảo dữ liệu được hợp lệ hóa và cập nhật chính xác.
* Cung cấp khả năng thống kê báo cáo linh hoạt.

**1.2. Hạn chế**

* Do khả năng còn hạn chế nên một số chức năng chưa được xây dựng: Tự động nhắc đơn hàng theo chu kỳ.

**1.3. Kinh nghiệm thu được**

* Củng cố thêm kiến thức đã học về các môn phân tích thiết kế hệ thống, cơ sở dữ liệu, thiết kế web, ứng dụng web,...
* Tích lũy và học hỏi các kinh nghiệm quý báu trong quản lý dự án.
* Nâng cao kĩ năng làm việc theo nhóm.

**2. Hướng phát triển**

* Xây dựng các chức năng chưa hoàn thiện
* Chỉnh sửa giao diện hợp lý hơn, sinh động hơn mà vẫn đảm bảo tốc độ.
* Bổ sung các tính năng phát sinh khi người dùng yêu cầu

**TÀI LIỆU THAM KHẢO**

[1]. “Giáo trình phân tích và thiết kế hệ thống thông tin”, Trường Đại học Sao Đỏ.

[2]. “Giáo trình cơ sở dữ liệu”, Trường Đại học Sao Đỏ.

[3]. “Giáo trình thiết kế web”, Trường Đại học Sao Đỏ.

[4]. “Giáo trình phát triển ứng dụng web”, Trường Đại học Sao Đỏ.

Một số trang web tham khảo

[1]. http://elearning.saodo.edu.vn/

[2]. https://www.w3schools.com/

[3]. https://vi.wikipedia.org

[4]. https://codeigniter.com/docs

[5]. https://freetuts.net/