**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

**Đề tài:**

**WEBSITE CHUYÊN**

**KINH DOANH TIỂU THUYẾT**

**Người hướng dẫn : ThS. NGUYỄN NGỌC DUY**

**Sinh viên thực hiện : VĂN ĐỨC THÀNH**

**Mã số sinh viên : N18DCCN207**

**Lớp : D18CQCP02-N**

**Khóa : 2018-2023**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**Tp.Hcm, Tháng 08/2022**

**LỜI CẢM ƠN**

Lời nói đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến các thầy cô trong Khoa Công nghệ Thông tin 2 đã tạo mọi điều kiện thuận lợi để em có thể thực hiện được bài báo cáo thực tập này. Đồng thời, em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Ngọc Duy đã phụ trách hướng dẫn trực tiếp cho em, đã không ngần ngại những khó khăn, vất vả để luôn tích cực hỗ trợ em hết mình.

Trong quá trình thực hiện bài viết báo cáo, nếu như có nhiều những thiếu sót, em xin các thầy cô trong khoa có thể tiếp tục hỗ trợ, đóng góp ý kiến, phản hồi cũng như cung cấp thêm những hướng dẫn chi tiết cụ thể về mặt chuyên môn để em có thể thực hiện được nội dung được hoàn chỉnh nhất.

TP.HCM, tháng 08 năm 2022

Sinh viên thực hiện

Văn Đức Thành

**MỤC LỤC**

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 1](#_Toc113194353)

[1 Tổng quan: 1](#_Toc113194354)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT ĐỀ TÀI 2](#_Toc113194355)

[2.1 Xây dựng hệ thống 2](#_Toc113194356)

[2.1.1 Nền tảng công nghệ 2](#_Toc113194357)

[2.1.2 Công cụ lập trình 2](#_Toc113194358)

[2.2 Giới thiệu về Microsoft SQL Server Management 2](#_Toc113194359)

[2.3 Giới thiệu về ngôn ngữ lập trình Java 2](#_Toc113194360)

[2.3.1 Java là gì? 2](#_Toc113194361)

[2.3.2 Đặc điểm của Java 3](#_Toc113194362)

[2.3.3 Máy ảo Java 5](#_Toc113194363)

[2.3.4 Các phiên bản của Java 6](#_Toc113194364)

[2.4 Giới thiệu thư viện hỗ trợ 7](#_Toc113194365)

[2.4.1 Spring framework 7](#_Toc113194366)

[2.4.2 Bootstrap 13](#_Toc113194367)

[2.5 Tổng quan về RESTful API 14](#_Toc113194368)

[CHƯƠNG III: PHÂN TÍCH THIẾT KẾ VÀ CƠ SỞ DỮ LIỆU 15](#_Toc113194369)

[3.1 Phân tích thiết kế 15](#_Toc113194370)

[3.1.1. Yêu cầu 15](#_Toc113194371)

[3.1.2. Sơ đồ Usecase 16](#_Toc113194372)

[3.2. Thiết lập mô hình thực thể ERD (ERD - Entity Relationship Diagram) 18](#_Toc113194373)

[3.2.1 Xác định các thực thể 18](#_Toc113194374)

[3.2.2 Mô hình thực thể kết hợp ERD (ERD - Entity Relationship Diagram 18](#_Toc113194375)

[3.2.3 Mô hình dữ liệu quan hệ ERD 19](#_Toc113194376)

[3.3 . Thiết kế dữ liệu 19](#_Toc113194377)

[3.3.1. Mô hình Diagram 19](#_Toc113194378)

[3.4 Từ điển dữ liệu 20](#_Toc113194379)

[CHƯƠNG IV: CHƯƠNG TRÌNH 26](#_Toc113194380)

[4.1 API 26](#_Toc113194381)

[4.1.1 Cấu trúc API 26](#_Toc113194382)

[4.1.2 . REST API 27](#_Toc113194383)

[4.2 Giao diện 28](#_Toc113194384)

[4.2.1 Giao diện người dùng 28](#_Toc113194385)

[4.2.2 Giao diện admin 38](#_Toc113194386)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 44](#_Toc113194387)

[DANH MỤC TÀI LIỆU THAM KHẢO 45](#_Toc113194388)

**KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| API: Application Programming  Interface | Giao diện lập trình ứng dụng |
| App: Application | Ứng dụng |
| ERD: Entity Relationship Diagram | Mô hình thực thể kết hợp |
| HTTP: HyperText Transfer  Protocol | Giao thức truyền tải siêu văn bản |
| IDE: Integrated development  environment | Môi trường phát triển tích hợp |
| MVC: Model – View – Controller | Một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính |
| RDBMS: Relational Database  Management System | Hệ quản trị cơ sở dữ liệu quan hệ |
| REST: Representational State  Transfer | Một tiêu chuẩn dùng trong việc thiết kế  API cho các ứng dụng web |
| JVM**:** Java Virtual Machine | 1 máy ảo **java** – dùng để thực thi các chương trình Java hay hiểu nôm na là trình thông dịch của Java |
|  |  |

**DANH MỤC BẢNG, HÌNH ẢNH**

[Hình 2.1. Mô hình JVM 6](#_Toc113194568)

[Hình 2.2. Các phiên bản của Java 6](#_Toc113194569)

[Hình 2.3. Kiến trúc tổng thể của Spring Framework 8](#_Toc113194570)

[Hình 3.1: Sơ đồ Usecase 16](#_Toc113194571)

[Hình 3.2. Mô hình ERD 18](#_Toc113194572)

Hình [3.3. Mô hình Diagram 19](#_Toc113194573)

[Hình 4.1. Mô hình ba lớp 26](#_Toc113194574)

[Hình 4.2. Cấu trúc thư mục API 27](#_Toc113194575)

[Hình 4.3 Màn hình đăng nhập cho khách hàng 28](#_Toc113194576)

[Hình 4.4. Màn hình đăng ký cho khách hàng 29](#_Toc113194577)

[Hình 4.5. Màn hình quên mật khẩu cho khách hàng 29](#_Toc113194578)

[Hình 4.6. Mail gửi lại mật khẩu mới cho khách hàng 30](#_Toc113194579)

[Hình 4.7 Màn hình header của trang web 30](#_Toc113194580)

[Hình 4.8 Màn hình footer của trang web 31](#_Toc113194581)

[Hình 4.9 Màn hình hiển thị sản phẩm 31](#_Toc113194582)

[Hình 4.10 Màn hình hiển thị sản phẩm được tìm kiếm 32](#_Toc113194583)

[Hình 4.11 Màn hình hiển thị chi tiết sản phẩm 33](#_Toc113194584)

[Hình 4.12 Màn hình giỏ hàng 34](#_Toc113194585)

[Hình 4.13 Màn hình nhập thông tin khách hàng để nhận hàng 35](#_Toc113194586)

[Hình 4.14 Màn hình thanh toán online 35](#_Toc113194587)

[Hình 4.15 Màn hình lịch sử mua hàng 36](#_Toc113194588)

[Hình 4.16 Màn hình đổi mật khẩu 37](#_Toc113194589)

[Hình 4.17 Màn hình thay đổi thông tin cá nhân 37](#_Toc113194590)

[Hình 4.18 Màn hình quản lí đơn hàng 38](#_Toc113194591)

[Hình 4.19 Màn hình quản lí sản phẩm 39](#_Toc113194592)

[Hình 4.20 Màn hình thêm sản phẩm 39](#_Toc113194593)

[Hình 4.21 Màn hình sửa sản phẩm 40](#_Toc113194594)

[Hình 4.22 Màn hình danh sách khách hàng 40](#_Toc113194595)

[Hình 4.23 Màn hình quản trị danh mục sản phẩm 41](#_Toc113194596)

[Hình 4.24 Màn hình thống kê sản phẩm bán chạy 41](#_Toc113194597)

[Hình 4.25 Màn hình thống kê doanh thu theo ngày 42](#_Toc113194598)

[Hình 4.26 Màn hình thống kê doanh thu theo tháng 43](#_Toc113194599)

[Hình 4.27 Màn hình thống kê doanh thu theo năm 43](#_Toc113194600)

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

## 1 Tổng quan:

Cùng với sự phát triển của nền kinh tế, nhu cầu mua sắm trong xã hội không ngừng tăng lên và ngày càng trở nên phong phú, đa dạng hơn. Trong xã hội xuất hiện rất nhiều người thích đọc sách, truyện như tiểu thuyết , đồng thời lại có một số người có sẵn sản phẩm và muốn nhanh chóng bán nó. Ban đầu những người này thường tìm gặp nhau một cách trực tiếp để trao đổi mua bán, nhưng khi cung và cầu vốn không ngừng tăng lên thì hình thức mua bán trực tiếp không đáp ứng được nữa, do đó đòi hỏi một thị trường cung cấp sự mua bán nhanh chóng, đáp ứng nhu cầu tài chính của nhau. Đó là hệ thống mua bán online.

Trong lịch sử hình thành và phát triển của internet, thị trường online được hình thành như nhu cầu tất yếu để thuận tiện trong việc cung và cầu sản phẩm. Sau đó, cùng với sự phát triển của nền kinh tế, nhu cầu mua sắm cũng tăng lên rất nhiều và những người thích đọc sách cũng ngày một tăng, đặc biệt bây giờ, giới trẻ rất thích đọc truyện, tiểu thuyết nên việc tạo ra một trang web để các độc giả có thể thuận tiện mua sắm các quyển tiểu thuyết mình thích cũng như ủng hộ tác giả một phần tiền để cho ngành viết tiểu thuyết ngày một phát triển và đặc biệt là để giúp mọi người có thể ủng hộ sự bản quyền, công sức của những tác giả viết ra tiểu thuyết.

Bằng việc thực hiện đề tài “Website chuyên kinh doanh tiểu thuyết”, giúp cho các nhà xuất bản, tác giả có thể tiêu thụ được tiểu thuyết mình sáng tác, in ấn ra. Hiểu được cơ bản về vấn bản quyền, thuận tiện mua sắm tại chỗ, nhanh chóng. Từ đó việc tạo ra nhưng website về việc kinh doanh tiểu thuyết bản quyền là một việc rất cần thiết và hợp lí để giúp ngành sáng tác tiểu thuyết ngày một phát triển.

# 

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT ĐỀ TÀI

## Xây dựng hệ thống

## 2.1.1 Nền tảng công nghệ

- Web : sử dụng công nghệ Spring MVC kết hợp Thymeleaf và REST Template.

- RESTful WebService: Sử dụng framework Spring Boot.

### 2.1.2 Công cụ lập trình

- Hệ quản trị cơ sở dữ liệu: SQL Server

- Trình biên dịch/ công cụ lập trình: IDE Eclipse

- Ngôn ngữ sử dụng: Java

- Công nghệ xây dựng Front-end: HTML,CSS, Bootstrap, Javascript, Spring MVC

- Công nghệ xây dựng Back-end: Java Spring framework (Spring Boot, Spring data JPA, Spring Security)

## Giới thiệu về Microsoft SQL Server Management

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mạng máy tính hoạt động theo mô hình khách chủ cho phép đồng thời cùng lúc có nhiều người dùng truy xuất đến dữ liệu, quản lý việc truy nhập hợp lệ và các quyền hạn của từng người dùng trên mạng.

Microsoft SQL hỗ trợ sao lưu tự động dữ liệu qua JOB. Việc làm này có thể được tự động hóa một cách dễ dàng thông qua SQL Server Agent. Đây là module trong.

Microsoft SQL Server cung cấp các loại đồng bộ dữ liệu bao gồm Transactional replication, Merge replication và Snapshot replication nhằm mục đích đảm bảo an toàn dữ liệu và giải quyết bài toán nhiều người truy cập cùng một thời điểm.

## Giới thiệu về ngôn ngữ lập trình Java

### Java là gì?

* + - * Java là một trong những ngôn ngữ lập trình hướng đối tượng.nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.
      * Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.
      * Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.
      * Java được tạo ra với tiêu chí “Viết một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

### 2.3.2 Đặc điểm của Java

-   **Là một ngôn ngữ thuần hướng đối tượng**

Trong quá trình tạo ra một ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã kế thừa các cú pháp của C và C++.

Tuy nhiên, trong Java thao tác với con trỏ bị lược bỏ nhằm đảm bảo tính an toàn và dễ sử dụng hơn. Các thao tác overload, goto hay các cấu trúc như struct và union cũng được lược bỏ.

-   **Độc lập phần cứng và hệ điều hành**

Một chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng “cross-platform”. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ mã nguồn và cấp độ nhị phân.

Cấp mã nguồn: kiểu dữ liệu trong Java nhất quán cho tất cả hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi

Cấp nhị phân: một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có JVM (Java Virtual Machine) để thông dịch đoạn mã này.

-  **Ngôn ngữ thông dịch**

Ngôn ngữ lập trình thường được chia ra làm 2 loại (Tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch

Thông dịch: nó dịch từng lệnh rồi chạy từng lệnh, muốn chạy lại thì phải dịch lại.

Biên dịch: code sau khi được biên dịch sẽ tạo 1 file thường là .exe và file này có thể đem sử dụng lại mà không cần biên dịch

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể như sau

Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã byte code. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã bytecode này thành machine code  (hay native code) khi nhận được yêu cầu chạy chương trình.

Ưu điểm : Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.

Nhược điểm : Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được)

- **Đa luồng**

Java hỗ trợ lập trình đa tiến trình (multithread) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority…).

- **Tính an toàn và bảo mật**

*Tính an toàn*

Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu.

Dữ liệu phải được khai báo tường minh.

Không sử dụng con trỏ và các phép toán với con trỏ.

Java kiểm soát chặt chẽ việc truy nhập vào mảng, chuỗi. Không cho phép sử dụng các kỹ thuật tràn. Do đó các truy nhập sẽ không vượt quá kích thước của mảng và chuỗi.

Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động.

Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.

*Tính bảo mật*

Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau.

Mức 1: chỉ có thể truy xuất dữ liệu cũng như phương thức thông qua giao diện mà lớp cung cấp.

Mức 2: trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.

Mức 3: Trình thông dịch sẽ kiểm tra mã bytecode xem các đoạn mã này có đảm bảo được các quy định quy tắc trước khi thực thi.

Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

### 2.3.3 Máy ảo Java

Để đảm bảo tính đa nền, Java sử dụng cơ chế Máy ảo của Java.

ByteCode là ngôn ngữ máy của Máy ảo Java tương tự như các lệnh nhị phân của các máy tính thực.

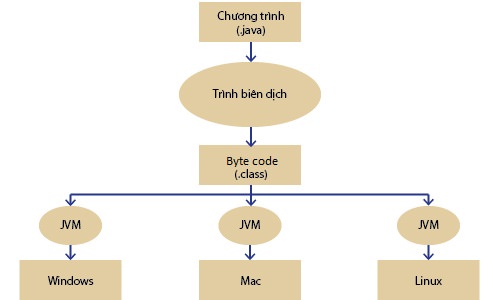
Một chương trình sau khi được viết bằng ngôn ngữ Java (có phần mở rộng là .java) phải được biên dịch thành tập tin thực thi được trên máy ảo Java (có phần mở rộng là .class).  Tập tin thực thi này chứa các chỉ thị dưới dạng mã Bytecode mà máy ảo Java hiểu được phải làm gì.

Khi thực hiện một chương trình, máy ảo Java lần lượt thông dịch các chỉ thị dưới dạng Bytecode thành các chỉ thị dạng nhị phân của máy tính thực và thực thi thực sự chúng trên máy tính thực (còn gọi là khả năng khả chuyển).

Máy ảo thực tế đó là một chương trình thông dịch. Vì thế các hệ điều hành khác nhau sẽ có các máy ảo khác nhau. Để thực thi một ứng dụng của Java trên một hệ điều hành cụ thể, cần phải cài đặt máy ảo tương ứng cho hệ điều hành đó.

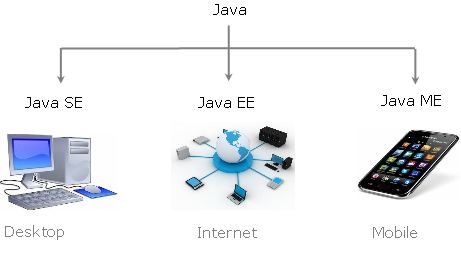
JVM cung cấp môi trường thực thi cho chương trình Java (còn gọi là khả năng độc lập với nền).

Có nhiều JVM cho các nền tảng khác nhau chẳng hạn như: Windows, Linux, và Mac.



Hình 2.1. Mô hình JVM

### 2.3.4 Các phiên bản của Java



Hình 2.2. Các phiên bản của Java

Java Standard Edition (Java SE) – Là một nền tảng cơ bản cho phép phát triển giao diện điều khiển, các ứng dụng mạng và các ứng dụng dạng Winform.

Java Enterprise Edition (Java EE) – Được xây dựng trên nền tảng Java SE, giúp phát triển các ứng dụng web, các ứng dụng ở cấp doanh nghiệp, …

Java Mobile Edition (Java ME) – Là một nền tảng cho phép phát triển các ứng dụng nhúng vào các thiết bị điện tử như mobile,…

## Giới thiệu thư viện hỗ trợ

### 2.4.1 Spring framework

* ***Spring là gì?***

Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lớn.

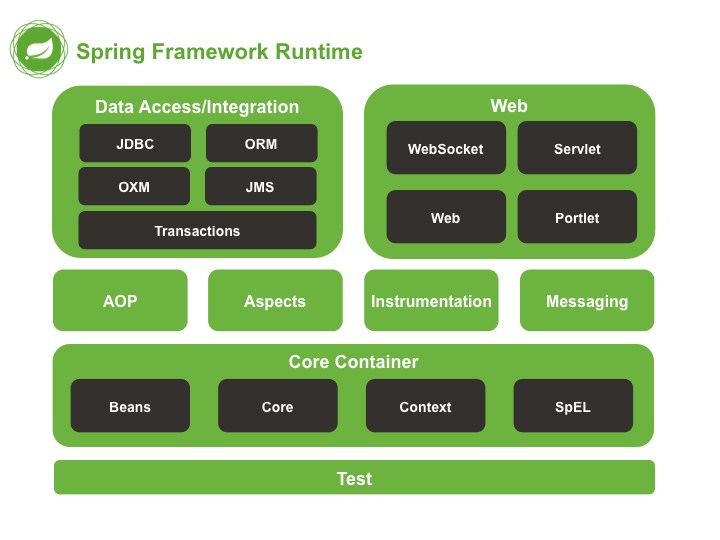
Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

* ***Kiến trúc, các module của Spring Framework***

Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó.

Dưới đây là kiến trúc tổng thể của Spring Framework.



Hình 2.3. Kiến trúc tổng thể của Spring Framework

**Test**

Tầng này cung cấp khả năng hỗ trợ kiểm thử với JUnit và TestNG.

**Spring Core Container**

Bao gồm các module spring core, beans, context và expression language (EL)

Spring core, bean cung cấp tính năng [IOC và Dependency Injection](https://stackjava.com/design-pattern/dependency-injection-di-la-gi.html).

Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX.

Expression Language được mở rộng từ Expression Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

**AOP, Aspects and Instrumentation**

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

**Data Access / Integration**

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database

**Web**

Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

Các lợi ích của Spring Framework

Spring cho phép các lập trình viên sử dụng POJOs và việc sử dụng POJOs này sẽ đảm bảo cho bạn không cần phải làm việc với EJB, các ứng dụng, các cấu hình và các luồng chạy,... chúng sẽ trở nên đơn giản hơn rất nhiều.

Spring thường được tổ chức dựa vào kiểu mô đun, nhờ vậy mà số lượng các lớp, các gói sẽ trở nên khá nhiều. Tuy nhiên, người dùng chỉ cần quan tâm đến những gì thì bạn cần mà những phần còn lại có thể bỏ qua.

Spring thường được sử dụng để hỗ trợ với nhiều loại công nghệ ORM Framework, JEE, các thư viện có khả năng tạo ra lập trình,...

Module Web của Spring đều được thiết kế dựa vào mô hình MVC nên nó có khả năng cung cấp rất đầy đủ những thay thế của các web [framework](https://itnavi.com.vn/blog/framework-java/) tương tự như Struts.

Web framework của Spring của một Web MVC framework có khả năng thiết kế tốt thì nó sẽ giúp bạn thay thế cho các Struts và những công nghệ kém phổ biến hơn.

Spring có khả năng cung cấp một API thuận tiện cho việc dịch ra được các ngoại lệ công nghệ cụ thể vào những trường hợp không được kiểm soát và có ngoại lệ nhất quán.

IoC Container sở hữu trọng lượng nhẹ nên rất có lợi cho quá trình phát triển cũng như triển khai những ứng dụng mới trên máy tính có bộ nhớ và tài nguyên CPU bị hạn chế.

Spring sẽ cung cấp được một giao diện quản lý transaction nhất quán. Từ đó, có thể mở rộng được đến một local transaction cũng như mở rộng lên các global transaction.

Thành phần cốt lõi của Spring Framework

Spring Core là một trong những thành phần quan trọng cốt lõi của Spring Framework. Đây là một trong những nền tảng được đánh giá bao nhờ có sự vững chắc giúp bạn có thể xây dựng được nhiều thành phần khác từ các hệ sinh thái Spring Framework. Những thành phần hỗ trợ cho Spring Core như sau:

**Spring Bean**

Spring Bean được xem là trung tâm của Spring Core và  là một trong những trái tim ứng dụng của Spring. Bởi vì Spring Framework được thiết kế bằng cách sử dụng các POJO hoặc các Spring Bean và bạn có thể hiểu Spring Bean là những đối tượng Java tương đối đơn giản.

**Dependency Injection (DI)**

Dependency Injection là một trong những sức mạnh vô cùng nổi bật của Spring Framework. Trong một số phiên bản gốc thì EJB người ta thường không nhận thấy sự xuất hiện của các Dependency Injection.

Tuy nhiên, do nhận thấy được sự thành công từ Spring nên các phiên bản EJB 3.x cũng đã cung cấp các annotation để bạn có thể thực hiện inject các dependency. Dependency Injection được đánh giá là một trong những mẫu thiết kế phần mềm đối tượng phụ thuộc thì sẽ được inject vào một lớp nào đó.

**Spring Context**

Spring Context giúp bạn mang mọi thứ lại với nhau và Spring Context thường được kế thừa từ những tính năng từ Spring Bean. Từ đó, bổ sung và hỗ trợ cho các internationalization (ví dụ như các resource bundle), event propagation, resource loading...

Ngoài ra, Spring Context có khả năng hỗ trợ các tính năng mới của Java như EE hoặc EJB, JMX,... Trong đó, Interface Application Context mới chính là một trong những trọng tâm quan trọng của Spring Context.

**Spring Expression Language (SpEL)**

Spring Expression Language là loại ngôn ngữ ngắn gọn có thể giúp cho các cấu hình Spring Framework trở nên linh hoạt và dễ hiểu hơn.

Các loại Spring phổ biến ngày nay

**Spring MVC**

Spring MVC là một trong những mô hình ứng dụng mà ở đó các thành phần được phân tách thành nhiều lớp riêng biệt và mỗi cái có các nhiệm vụ đặc trưng. Do nhu cầu xây dựng các ứng dụng trên nền tảng Web được nhanh chóng và tiện lợi nên Spring MVC đã ra đời giúp việc triển khai các dự án nhanh hơn.Spring MVC được xây dựng theo cấu trúc MVC (Model-View-Controller) ở đó có các API cho phép việc xây dựng các ứng dụng trên web được dễ dàng và có quy tắc hơn.

Cấu trúc Spring MVC và các thành phần chứa bên trong Model - View - Controller

**Model** là lớp chứa thông tin của các đối tượng (Dữ liệu), tương tác với Database. Trong Spring MVC nó gồm các file POJO, Service, DAO thực hiện truy cập database, xử lý business.

**View** là nơi hiển thị nội dung và tương tác trực tiếp với người dùng. Trên thực tế khi làm việc với Spring thì trong phần view người ra sẽ sử dụng các template engine mã nguồn mở cho phép chúng ta sử dụng lại các thành phần chung giữa các trang trong ứng dụng web và xử lý nhiều loại ngôn ngữ khác như HTML, XML, Javascript, CSS.

**Controller** là nơi tiếp nhận những request từ phía người dùng sau đó xử lý request để xây dựng dữ liệu cho view (model) và chọn view để trả lại kết quả của cho người dùng. Cụ thể ở đây là Dispatcher Controller, Handler Mapping, Controller.Thành phần chính của controller trong Spring MVC là:

1. Một annotation **@Controller** được khai báo cùng với định nghĩa của lớp **HomeController** và một annotation **@RequestMapping** được khai báo cùng với định nghĩa của phương thức **home().**
2. Phương thức **home()** trong lớp **HomeController** đang trả về một chuỗi **“home”**, chuỗi này định nghĩa tên view sẽ được dùng để hiển thị kết quả request cho người dùng. Biến **model** lưu trữ những dữ liệu cần thiết cho view **“home”** sử dụng và biến **locale** của đối tượng **Locale** để lấy dữ liệu về ngày giờ của server đang chạy web của chúng ta.

**Spring Boot**

Spring Boot là một Spring framework hiện đang được nhiều lập trình viên lựa chọn vì nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng. Spring Boot được coi như là điểm khởi đầu để xây dựng tất cả các ứng dụng dựa trên nền tảng Spring.

Bất cứ thứ gì sinh ra và ngày càng phát triển đều có lý do của nó, không phải tự nhiên mà Spring Boot lại được dùng để tạo các dự án Spring:

* Giúp xây dựng ứng dụng Spring độc lập (stand-alone) bao gồm gần như mọi thứ REST API, WebSocket, Web, Streaming …
* Giúp giảm tối ưu công đoạn cấu hình và không yêu cầu cấu hình thông qua XML.
* Có thể nhúng trực tiếp Tomcat, Jetty, Undertow… do đó không cần phải triển khai tập tin WAR.
* Tích hợp các loại mô-đun khác nhau trong các loại Spring như Spring Data, Spring MVC, Spring Security, Spring JDBC, Spring ORM…
* Các starter dependency làm cho việc thêm các hỗ trợ phù hợp cho project giúp cấu hình ban đầu nhanh hơn.
* Bạn có thể dễ dàng phát triển các ứng dụng Spring dựa trên các loại ngôn ngữ khác như Java hoặc Groovy.
* Spring Boot có thể ự động cấu hình Spring mỗi khi cần thiết.
* Hỗ trợ nhiều IDE như Spring Tool Suite, IntelliJ IDEA, NetBeans hoặc cũng có thể cấu hình đơn giản và dùng Sublime Text để phát triển luôn.
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …

**Spring Security**

Spring security là 1 framework thuộc hệ thống Spring, dành riêng cho việc thiết lập bảo mật của ứng dụng bao gồm authentication và authorization. Mặc dù hiện nay có rất nhiều bài hướng dẫn xây dựng một application sử dụng spring-security step by step, nhưng vẫn còn một số khái niệm dễ gây nhầm lẫn, nhất là với những bạn bắt đầu làm quen nên mình muốn tổng hợp lại trong bài viết này.

**Spring Batch, Spring IO, Spring Cloud, ….**

### 2.4.2 Bootstrap

* Bootstrap là gì?

Bootstrap là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,...

Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế responsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

* Tại sao chúng ta nên sử dụng Bootstrap?

Bootstrap là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng nên một website. Bootstrap đã xây dựng nên 1 chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, chúng ta hay nghe tới một cụm từ rất thông dụng "Thiết kế theo chuẩn Bootstrap".

Từ cái "chuẩn mực" này, chúng ta có thể thấy rõ được những điểm thuận lợi khi sử dụng Bootstrap.

Rất dễ để sử dụng: Nó đơn giản vì nó được thực hiện dựa trên HTML, CSS và Javascript chỉ cần có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.

Responsive: Bootstrap xây dựng sẵn responsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.

Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

* Cách sử dụng

Có 2 cách để sử dụng Bootstrap

1. Download Bootstrap packet từ getbootstrap.com
2. Thêm Bootstrap từ CDN

<!-- Latest compiled and minified CSS →

<link rel="stylesheet"

href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css">

<!-- Optional theme -->

<link rel="stylesheet"

href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->

<script src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>

## Tổng quan về RESTful API

REST là viết tắt của Representational State Transfer. RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

# CHƯƠNG III: PHÂN TÍCH THIẾT KẾ VÀ CƠ SỞ DỮ LIỆU

## 3.1 Phân tích thiết kế

## 3.1.1. Yêu cầu

**a) Yêu cầu chức năng hệ thống**

**Quản trị:**

- Xác nhận đơn hàng (chuyển trạng thái đơn hàng)

- Thêm/ xóa/ sửa sản phẩm

- Thống kê doanh số

- Quản lí khuyến mãi

**Người dùng**

- Đăng ký/ đăng nhập tài khoản

- Cập nhật thông tin tài khoản

- Xem thông tin sản phẩm

- Thêm vào giỏ hàng

- Chọn mua sản phẩm

- Thanh toán

- Lịch sử mua hàng

**Đối tượng tham gia hệ thống**

- Nhân viên quản lý

- Người tiêu dùng

**b) Yêu cầu phi chức năng**

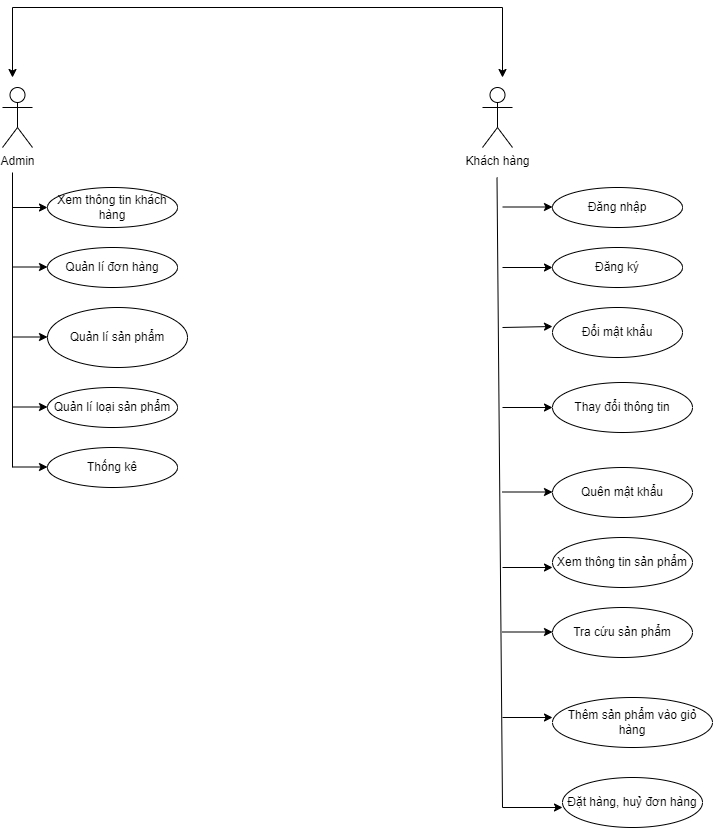
**-** Hệ thống tin cậy, dễ sử dụng, dễ thao tác, giao diện đẹp

- Đảm bảo tính bảo mật cho người dùng.

- Validate dữ liệu đầu vào.

- Có tính thực dụng.

### 3.1.2. Sơ đồ Usecase

****

Hình 3.1: Sơ đồ Usecase

|  |  |  |
| --- | --- | --- |
| STT | Tên Usecase | Ý nghĩa/ Ghi chú |
| 1 | Đăng nhập | Người dùng cần đăng nhập trước khi thực hiện thao tác với giỏ hàng |
| 2 | Đăng ký | Người dùng cần đăng kí trước khi thực hiện đăng nhập |
| 3 | Đổi mật khẩu | Người dùng đổi mật khẩu của tài khoản cá nhân |
| 4 | Thay đổi thông tin | Người dùng thay đổi thôg tin tài khoản |
| 5 | Quên mật khẩu | Người dùng lấy lại mật khẩu khi quên mật khẩu |
| 6 | Xem thông tin sản phẩm | Người dùng xem thông tin chi tiết về sản phẩm |
| 7 | Tra cứu sản phẩm | Người dùng tìm kiếm sản phẩm qua các từ khoá |
| 8 | Thêm sản phẩm vào giỏ hàng | Người dùng thêm sản phẩm vào giỏ hàng và thao tác trong giỏ hàng |
| 9 | Đặt – Huỷ đơn hàng | Người dùng đặt hàng và huỷ đơn hàng khi chưa được admin xác nhận đơn hàng |
| 10 | Xem thông tin khách hàng | Admin xem thông tin các khách hàng |
| 11 | Quản lí đơn hàng | Admin quản lí trạng thái đơn hàng ( xác nhận, huỷ đơn, xác nhận thanh toán) |
| 12 | Quản lí sản phẩm | Admin thêm, xoá , sửa sản phẩm |
| 13 | Quản lí loại sản phẩm | Admin thêm, sửa loại sản phẩm |
| 14 | Thống kê | Thống kê sản phẩm bán chạy |

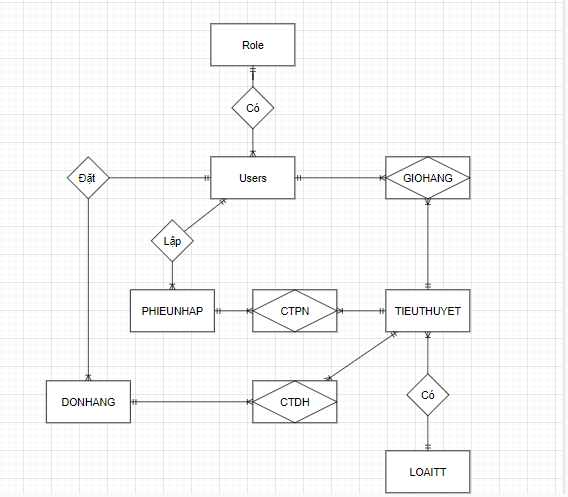
Bảng 3.1: Mô tả Usecase

## 3.2. Thiết lập mô hình thực thể ERD (ERD - Entity Relationship Diagram)

### 3.2.1 Xác định các thực thể

* + - * **PhieuNhap** (MaPN, TrangThai,**MaNV**)
      * **CTPN** (MaPN, **MaSach**, SoLuong, DonGia)
      * **TieuThuyet** (MaSach, TenSach, SoLuong, MoTa, TacGia, DonGia,GiamGia, HinhAnh, NXN, **LoaiTT\_ID**)
      * **LoaiTT** (LoaiTT\_ID, TenLoaiTT)
      * **GioHang** (IDGH, **MAKH**, **MASACH**, SoLuong)
      * **DonHang** (IDDH, NgayLap, TrangThai, **MaKH**, SDT, DiaChi, TongGia,GhiChu)
      * **CTDH** (IDCTDH, **IDDH**, **MaSach**, SoLuong, DonGia)
      * **NhanVien** (MaNV, HoTen, DiaChi, SDT, PassWord, **RoleID**, Email )
      * **Role** (RoleID, RoleName )
      * **KhachHang** (MaKH, HoTen, DiaChi, SDT, PassWord, Email)

### *Mô hình thực thể kết hợp ERD (ERD - Entity Relationship Diagram*

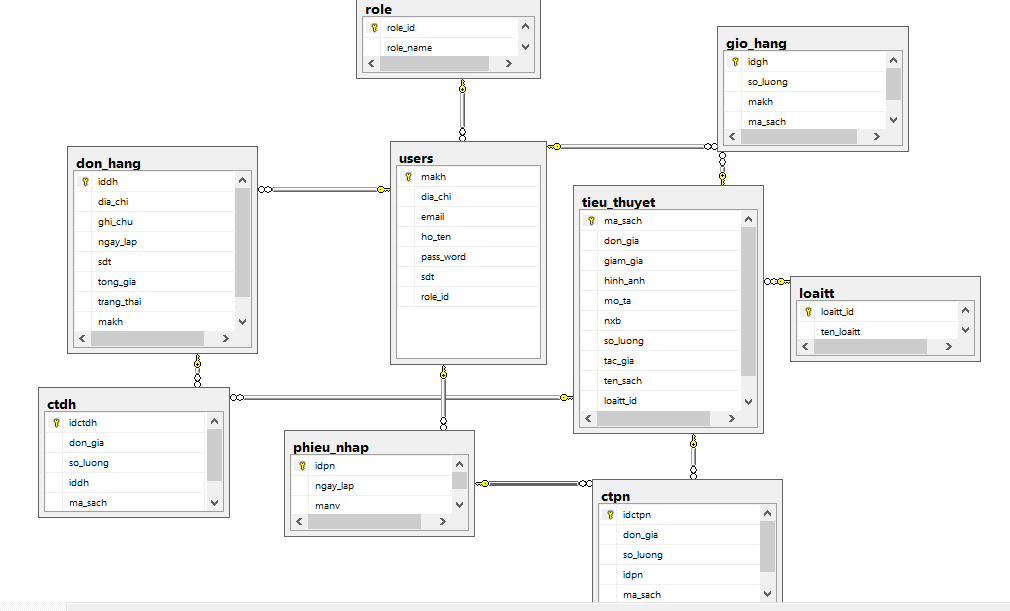


Hình 3.2. Mô hình ERD

### *Mô hình dữ liệu quan hệ ERD*

* + - * **PhieuNhap** (MaPN, TrangThai,**MaNV**)
      * **CTPN** (MaPN, **MaSach**, SoLuong, DonGia)
      * **TieuThuyet** (MaSach, TenSach, SoLuong, MoTa, TacGia, DonGia,GiamGia, HinhAnh, NXN, **LoaiTT\_ID**)
      * **LoaiTT** (LoaiTT\_ID, TenLoaiTT)
      * **GioHang** (IDGH, **MAKH**, **MASACH**, SoLuong)
      * **DonHang** (IDDH, NgayLap, TrangThai, **MaKH**, SDT, DiaChi, TongGia,GhiChu)
      * **CTDH** (IDCTDH, **IDDH**, **MaSach**, SoLuong, DonGia)
      * **NhanVien** (MaNV, HoTen, DiaChi, SDT, PassWord, **RoleID**, Email )
      * **Role** (RoleID, RoleName )
      * **KhachHang** (MAKH, HoTen, DiaChi, SDT, PassWord, Email)
  1. **. Thiết kế dữ liệu**

**3.3.1. Mô hình Diagram**



3.3. Mô hình Diagram

* 1. **Từ điển dữ liệu**
* Bảng: **KhachHang** (maKH, email, password, hoten, sdt, diachi)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | maKH | Bigint | Id tài khoản |  | X |  | X |  |
| 2 | Email | nvarchar(255) | Email người dùng | X |  |  | X |  |
| 3 | Password | nvarchar(255) | Mật khẩu |  |  |  | X |  |
| 4 | hoten | nvarchar(255) | Tên người dùng |  |  |  | X |  |
| 5 | sdt | nvarchar(11) | Số điện thoại |  |  |  |  |  |
| 6 | diachi | nvarchar(255) | Địa chỉ |  |  |  |  |  |

* Bảng: **NhanVien** (maKH, email, password, hoten, sdt, diachi, **role\_id**)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | maNV | Bigint | Id tài khoản |  | X |  | X |  |
| 2 | Email | nvarchar(255) | Email người dùng | X |  |  | X |  |
| 3 | Password | nvarchar(255) | Mật khẩu |  |  |  | X |  |
| 4 | hoten | nvarchar(255) | Tên người dùng |  |  |  | X |  |
| 5 | sdt | nvarchar(11) | Số điện thoại |  |  |  |  |  |
| 6 | diachi | nvarchar(255) | Địa chỉ |  |  |  |  |  |
| 7 | role\_id | Bigint | Id quyền |  |  | X | X |  |

* Bảng: **Role**( role\_id, role\_name )

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | role\_id | Bigint | Id chức vụ |  | X |  | X |  |
| 2 | role\_name | Nvarchar  (255) | Tên chức vụ | X |  |  | X |  |

* Bảng: **LoaiTT**( loaitt\_id, tenloaitt )

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | loaitt\_id | Bigint | Id danh mục |  | X |  | X |  |
| 2 | tenloaitt | Nvarchar  (255) | Tên danh mục | X |  |  | X |  |

* Bảng: **TieuThuyet** (maSach,tenSach**,** moTa**,** donGia, giamGia, soLuong, hinhAnh, nxb, **loaitt\_id**)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | maSach | Bigint | Id sản phẩm |  | X |  | X |  |
| 2 | tenSach | nvarchar(255) | Tên sản phẩm |  |  |  | X |  |
| 3 | moTa | nvarchar(MAX) | Mô tả chi tiết sản phẩm |  |  |  | X |  |
| 4 | donGia | int | Giá sản phẩm |  |  |  | X |  |
| 5 | giamGia | int | Giảm giá |  |  |  |  |  |
| 5 | soLuong | int | Số lượng tồn |  |  |  | X | 0 |
| 6 | hinhAnh | nvarchar(MAX) | Hình ảnh sản phẩm |  |  |  | X |  |
| 7 | nxb | nvarchar(255) | Nhà Xuất Bản |  |  |  | X |  |
| 8 | loaitt\_id | Bigint | Mã danh mục |  |  | X | X |  |

* Bảng: **DonHang** (iddh,**maKH,** ngayLap**,** diaChi, sdt, trangThai, ghiChu)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | iddh | Bigint | Id đơn hàng |  | X |  | X |  |
| 2 | maKH | Bigint | Id người dùng |  |  | X | X |  |
| 3 | ngayLap | DateTime(6) | Ngày tạo đơn hàng |  |  |  | X | getDate() |
| 4 | diaChi | nvarchar(255) | Địa chỉ nhận hàng |  |  |  | X |  |
| 5 | sdt | nvarchar(255) | SĐT nhận hàng |  |  |  | X |  |
| 6 | trangThai | Int | Trạng thái đơn hàng |  |  | X | X |  |
| 7 | ghiChu | nvarchar(255) | Ghi chú |  |  |  |  |  |
| 8 | tongGia | double | Tổng giá |  |  |  | X |  |

* Bảng: **CTDH** (idctdh,**iddh, maSach,** soLuong, donGia)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | idctdh | Bigint | Id CTDH |  | X | X | X |  |
| 2 | iddh | Bigint | Id đơn hàng |  |  | X | X |  |
| 3 | maSach | Bigint | Id sản phẩm |  |  | X | X |  |
| 4 | soLuong | int | Số lượng |  |  |  | X |  |
| 5 | donGia | int | Giá sản phẩm |  |  |  | X |  |

* Bảng: **GioHang**( idgh, **maKH, maSach**,soLuong, gia, hinhAnh)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | idgh | Bigint | Id giỏ hàng |  | X |  | X |  |
| 2 | maKH | Bigint | Id người dùng |  |  | X | X |  |
| 3 | maSach | Bigint | Id sản phẩm |  |  | X | X |  |
| 4 | soLuong | Int | Số lượng |  |  |  | X |  |

* Bảng: **PhieuNhap** (idpn**,** **maNV**, ngayLap)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | idpn | Bigint | Id phiếu nhập |  | X |  | X |  |
| 2 | maNV | Bigint | Id người dùng |  |  | X | X |  |
| 3 | ngayLap | DateTime(6) | Ngày tạo đơn |  |  |  | X | getDate() |

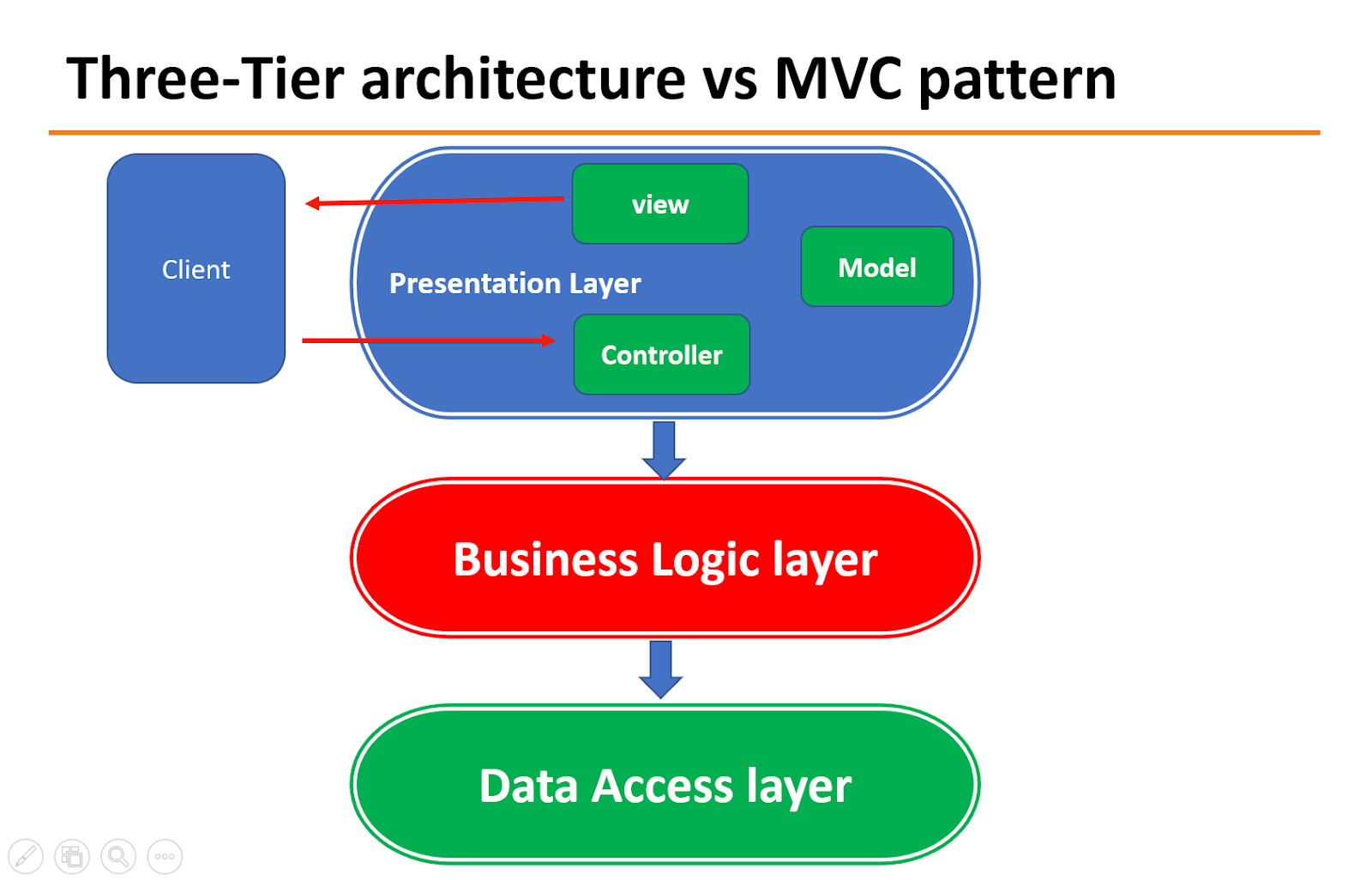
* Bảng: **CTPN** (idctpn,**idpn**, **maSach**, soLuong, donGia)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên Trường** | **Kiểu** | **Diễn giải** | **Unique** | **Primary Key** | **Foreign Key** | **Not Null** | **Default** |
| 1 | idctpn | Bigint | Id ctpn |  | X |  | X |  |
| 1 | idpn | Bigint | Id phiếu nhập |  |  | X | X |  |
| 2 | maSach | Bigint | Id sản phẩm |  |  | X | X |  |
| 3 | donGia | int | Giá sản phẩm |  |  |  | X |  |
| 4 | soLuong | Int | Số lượng |  |  |  | X |  |

# CHƯƠNG IV: CHƯƠNG TRÌNH

## 4.1 API

### *4.1.1 Cấu trúc API*



Hình 4.1. Mô hình ba lớp

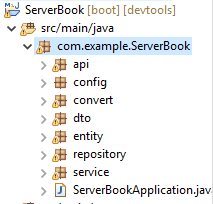
* + - * **Mô hình ba lớp (three tier)**

Đây là mô hình tổ chức source code rất phổ biến trong Spring Boot. Cụ thể, ứng dụng được chia làm 3 tầng (tier hoặc layer) như sau:

* **Presentation layer:** tầng này tương tác với người dùng, bằng View, Controller (trong MVC) hoặc API (nếu có).
* **Business logic layer:** Chứa toàn bộ logic của chương trình, các đa số code nằm ở đây
* **Data access layer:** Tương tác với database, trả về kết quả cho tầng business logic

Trong Spring Boot, thì có một số thành phần đại diện cho từng lớp:

* **Service:** chứa các business logic code
* **Repository:** đại diện cho tầng data access
  + - * **Cấu trúc thư mục**



Hình 4.2. Cấu trúc thư mục API

* + - API chứa các file tạo đường dẫn
    - Config để cấu hình cho ứng dụng server
    - Convert thực hiện việc chuyển đổi dữ liệu từ Entity sang dạng DTO
    - DTO là các class được custom lại từ Entity
    - Entity là các class thuộc tính theo từng table trong DB
    - Repository thực hiện việc kết nối DB
    - Service là tầng thực hiện các logic business

### 4.1.2 . REST API

REST viết tắt của REpresentational State Transfer. Đây là phong cách thiết kế hệ thống phân tán siêu phương tiện (distributed hypermedia systems) được giới thiệu đầu tiên năm 2000 bởi Roy Fielding trong nghiên cứu của mình.

- Resource: Nguyên tắc dựa trên các resource. Ví dụ: bài Post, thực thể Employee, thực thể Books…

- Sử dụng Id để định danh resource, ví dụ: /books/1 → Book có id = 1.

- Data format có thể nhiều dạng: Html, Json, XML,.... Tuỳ thuộc vào client yêu cầu

- Restful còn biết đến như là một phương thức phi trạng thái (Stateless). Điều đó có nghĩa là khác với ứng dụng Web thông thường, với Restful không có khái niệm về Session.

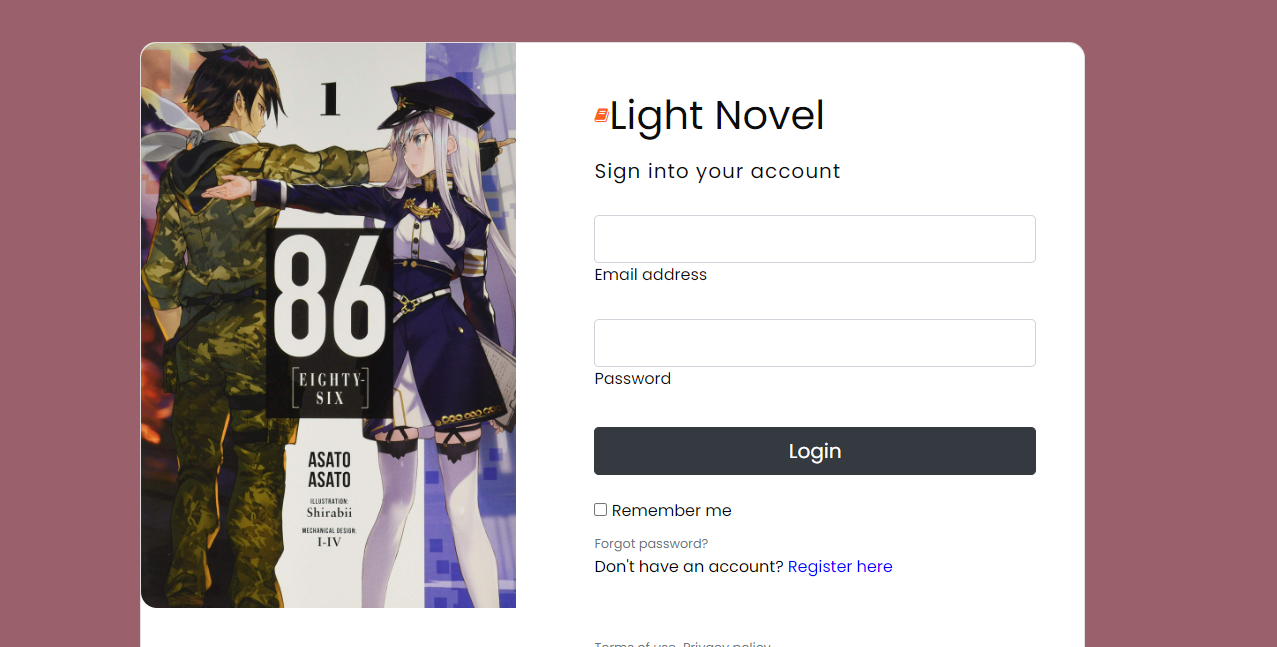
Resource Method: Dự trên giao thức của HTTP nên có các method sau:

* GET: Lấy dữ liệu
* POST: Tạo dữ liệu
* PUT: Tạo hoặc update dữ liệu
* DELETE: Xoá dữ liệu

## 4.2 Giao diện

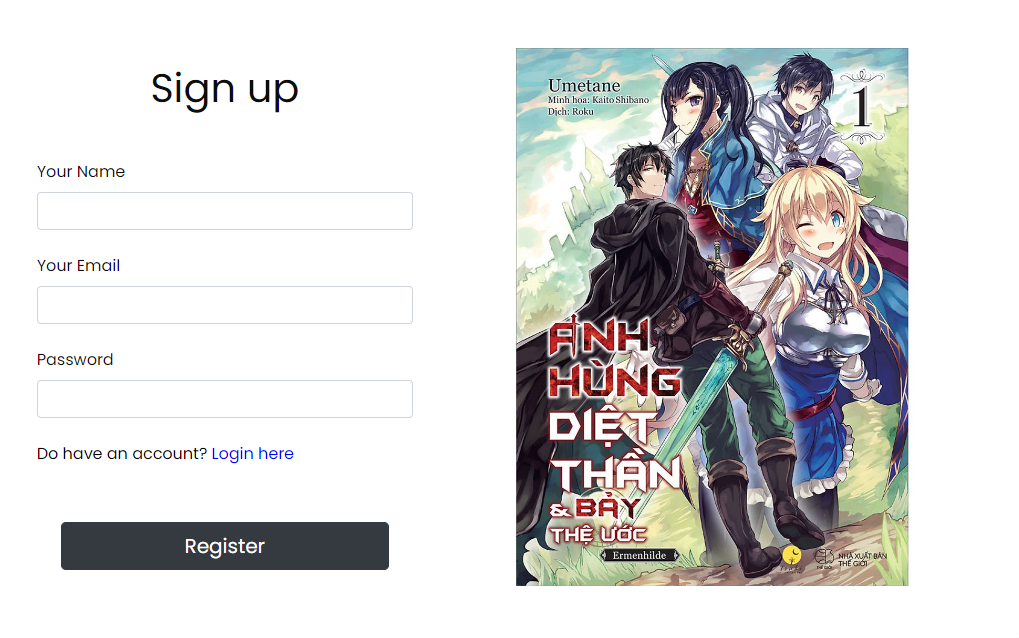
### 4.2.1 Giao diện người dùng

* **Màn hình đăng nhập cho khách hàng:**
* Khách hàng phải đăng nhập mới có thể mua hàng. Thông tin đăng nhập bao gồm email và mật khẩu. Nếu khách hàng chưa có tài khoản có thể đăng ký tài khoản.



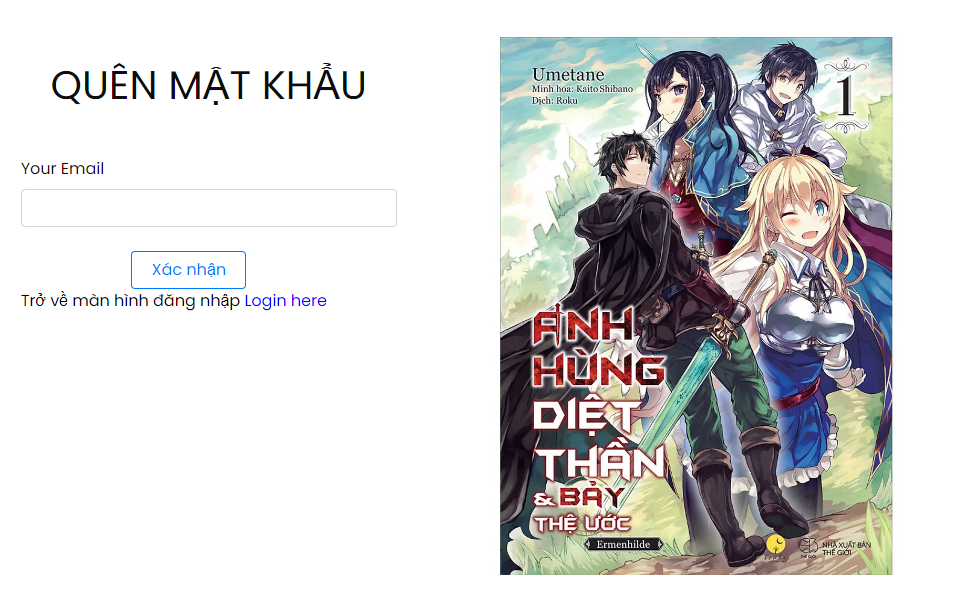
Hình 4.3 Màn hình đăng nhập cho khách hàng

* **Màn hình đăng kí cho khách hàng:**
* Khách hàng nhập tên, email và mật khẩu để đăng ký.
* Hệ thống sẽ kiểm tra email nhập vào nếu đã tồn tại sẽ báo đã tồn tại email. Nhập email chưa được khách hàng khác đăng kí sẽ đăng kí thành công.



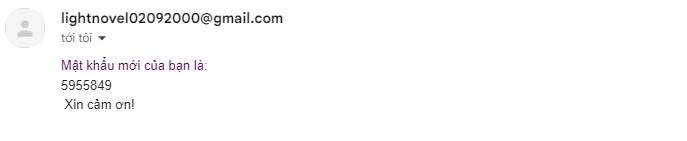
Hình 4.4. Màn hình đăng ký cho khách hàng

* **Màn hình quên mật khẩu cho khách hàng:**



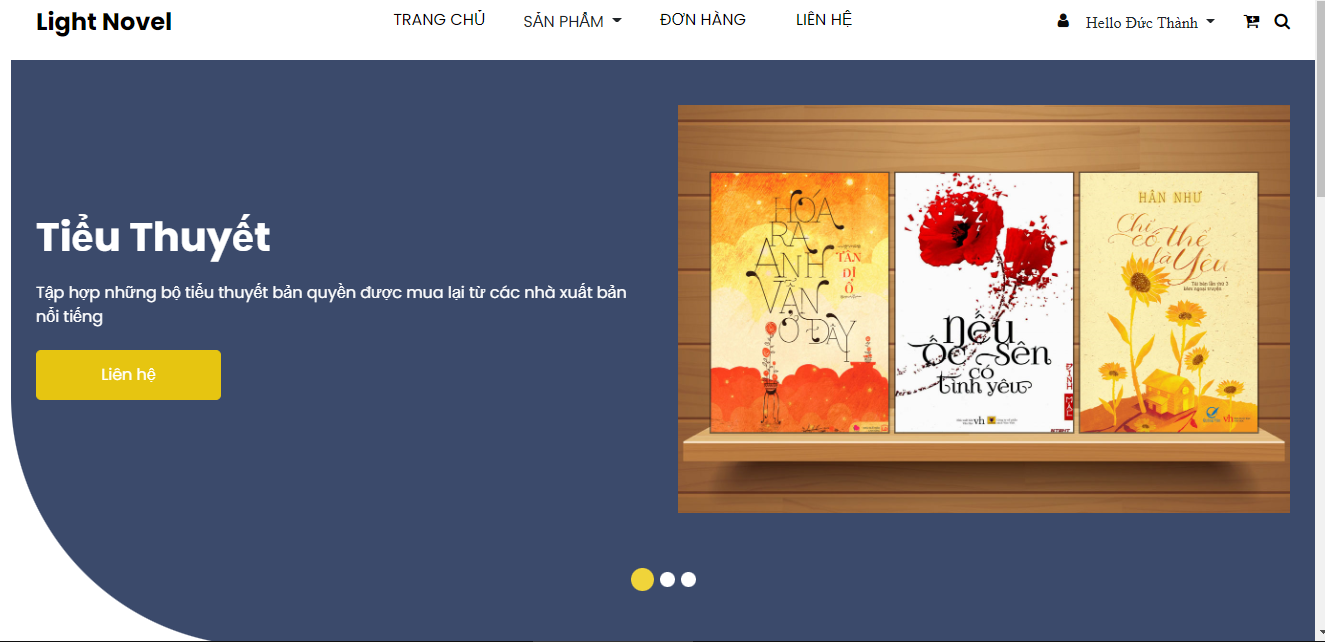
Hình 4.5. Màn hình quên mật khẩu cho khách hàng

* Khách hàng quên mật khẩu sẽ nhập email vào ô trống phía trên để hệ thống kiểm tra và gửi mật khẩu theo email đăng ký của khách hàng.

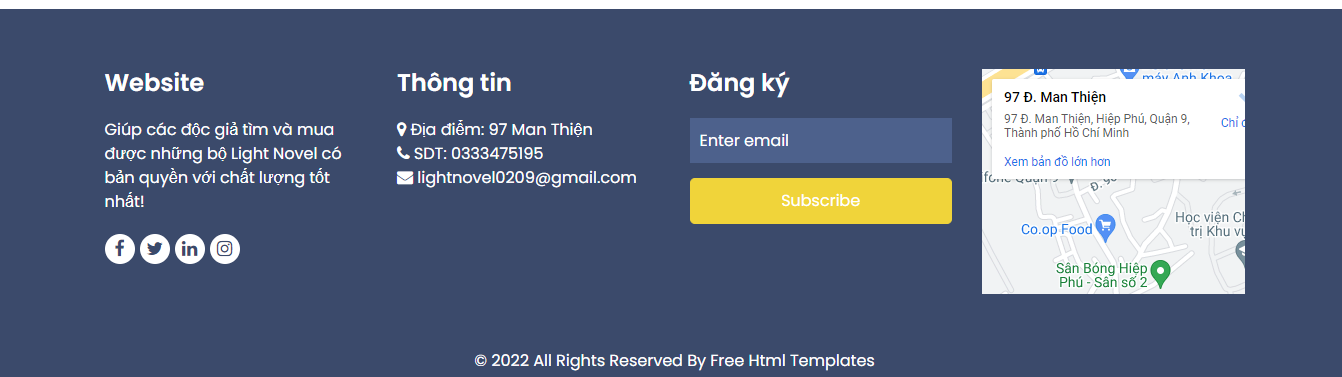


Hình 4.6. Mail gửi lại mật khẩu mới cho khách hàng

* Khách hàng sẽ xem gmail và lấy mật khẩu mới để đăng nhập vào website.
* **Màn hình trang chủ cho khách hàng:**

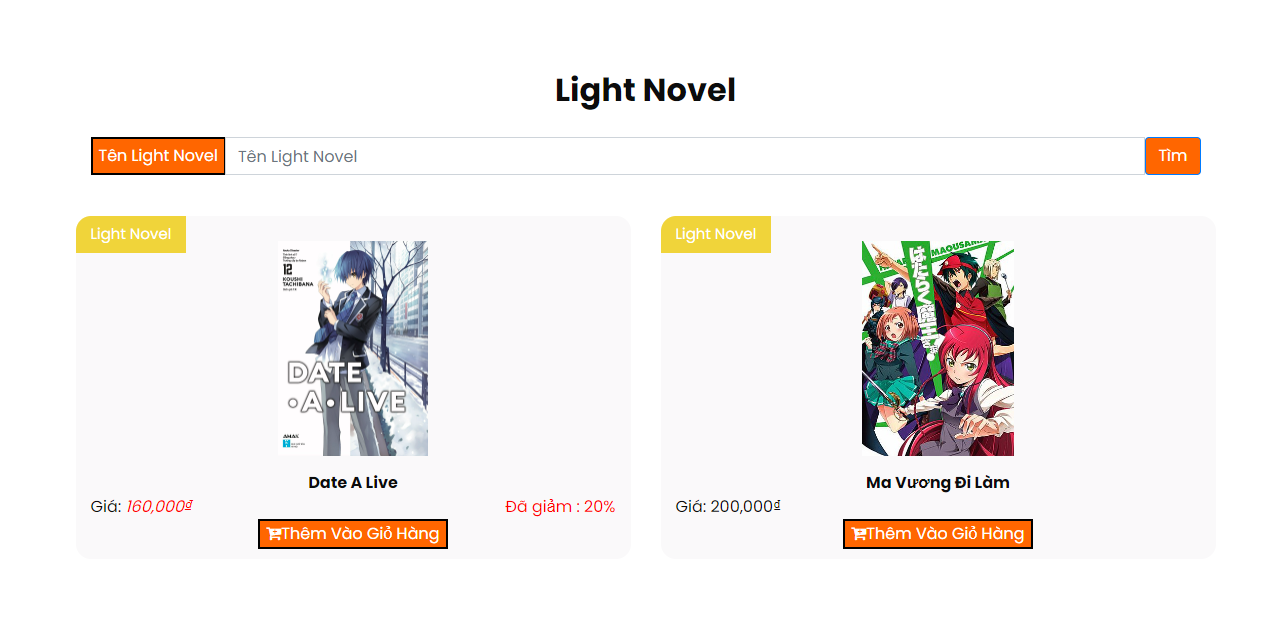


Hình 4.7 Màn hình header của trang web



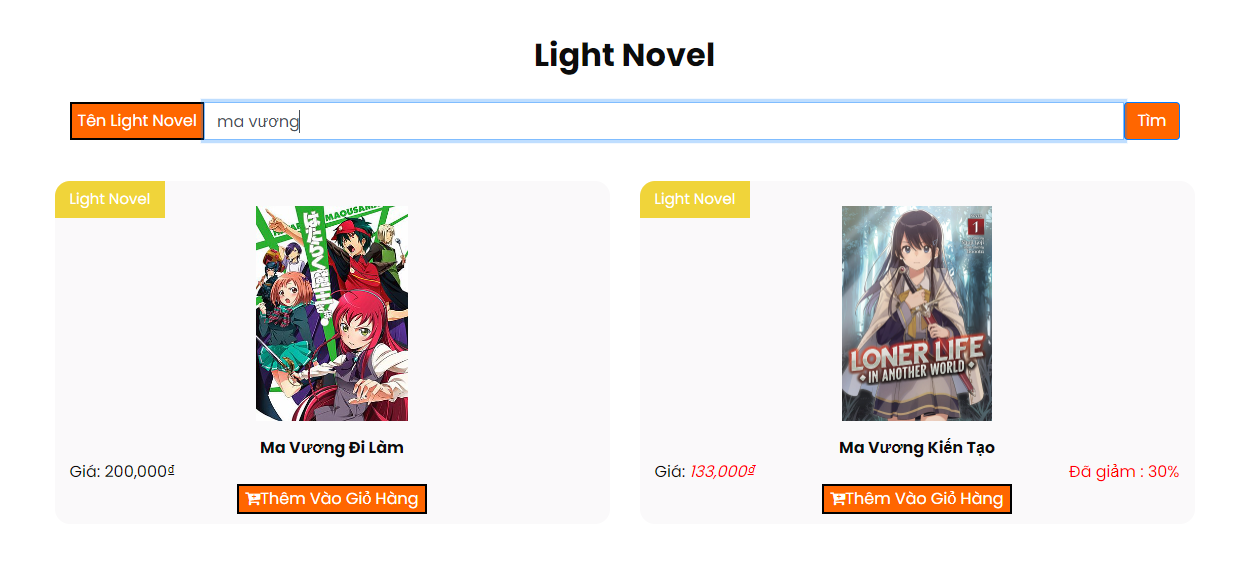
Hình 4.8 Màn hình footer của trang web

* **Màn hình hiển thị tất cả các sản phẩm hiện có:**
* Khách hàng có thể xem sản phẩm theo danh mục loại tiểu thuyết, hoặc có thể tìm kiếm sản phẩm theo tên hoặc mô tả. Thông tin sản phẩm có:
* Phần trăm khuyến mãi (nếu có).
* Hình ảnh của sản phẩm.
* Tên sản phẩm.
* Giá sản phẩm chưa áp dụng khuyến mãi
* Giá sản phẩm đã áp dụng khuyến mãi



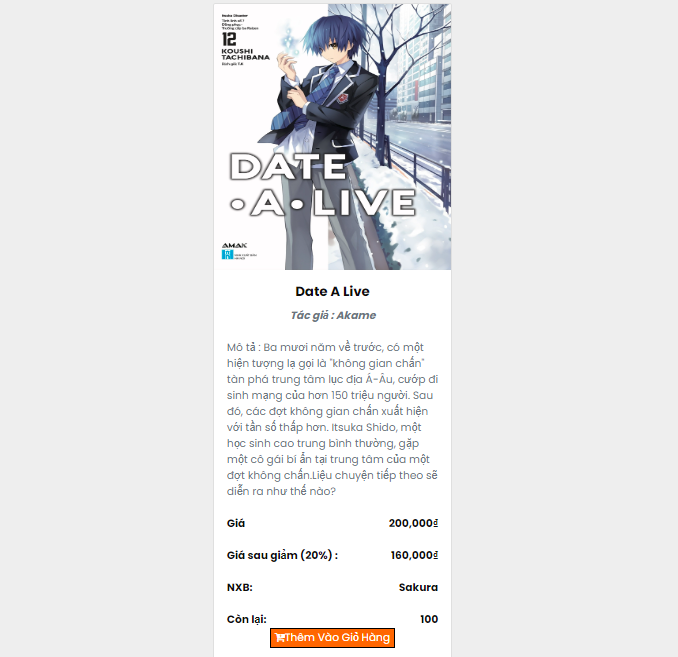
Hình 4.9 Màn hình hiển thị sản phẩm

* **Màn hình hiển thị sản phẩm tìm kiếm theo từ khoá :**
* Khách hàng có thể tìm kiếm sản phẩm bằng cách nhập từ khoá: phần mềm tìm kiếm theo tên không phân biệt ký tự nằm đầu, giữa hay cuối tên sản phẩm hoặc tìm kiếm theo mô tả của sản phẩm.



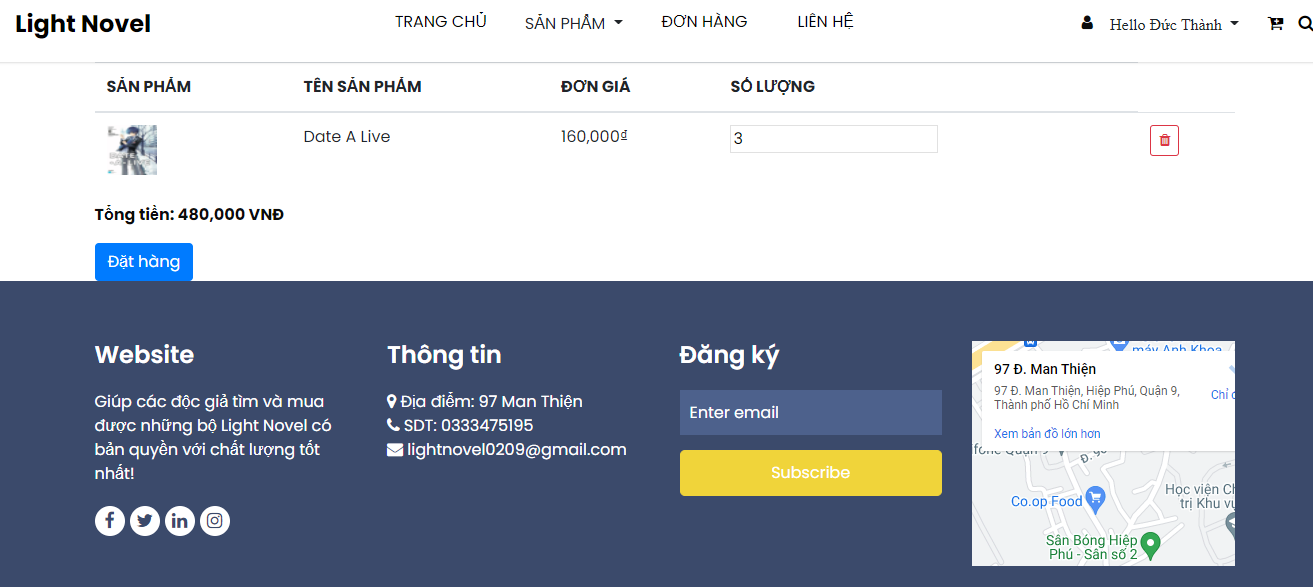
Hình 4.10 Màn hình hiển thị sản phẩm được tìm kiếm

* **Màn hình hiển thị thông tin chi tiết :**
* Khách hàng có thể click vào vùng ảnh của sản phẩm để hiển thị thông tin chi tiết sản phẩm.
* Khách hàng có thể xem chi tiết sản phẩm. Thông tin sản phẩm có:
* Hình ảnh của sản phẩm.
* Tên sản phẩm.
* Tên tác giả.
* Mô tả chi tiết về nội dung của tiểu thuyết.
* Giá sản phẩm
* Giá sản phẩm sau khi giảm
* Nhà xuất bản
* Số lượng sản phẩm còn lại
* Nút thêm vào giỏ hàng
* Khách hàng muốn thêm sản phẩm vào giỏ hàng sẽ nhấn nút “Thêm vào giỏ hàng”.
* Sản phẩm thêm vào giỏ hàng không thêm vào được nữa, chỉ có thể thêm sản phẩm khác.
* Khi đã thêm đủ sản phẩm thì click vào biểu tượng giỏ hàng để chuyển qua màn hình giỏ hàng.



Hình 4.11 Màn hình hiển thị chi tiết sản phẩm

* **Màn hình giỏ hàng:**
* Khách hàng xem thông tin giỏ hàng của mình. Thông tin giỏ hàng có:
* Thông tin sản phẩm: Tên, Hình ảnh, Giá đã áp dụng khuyến mãi.
* Số lượng sản phẩm.
* Tổng số tiền phải thanh toán.
* Khách hàng có thể tăng giảm số lượng sản phẩm, tối thiểu là một.
* Khi đã chọn số lượng vừa ý thì nhấn vào nút “Thanh toán”. Ứng dụng sẽ chuyển đến màn hình thanh toán.

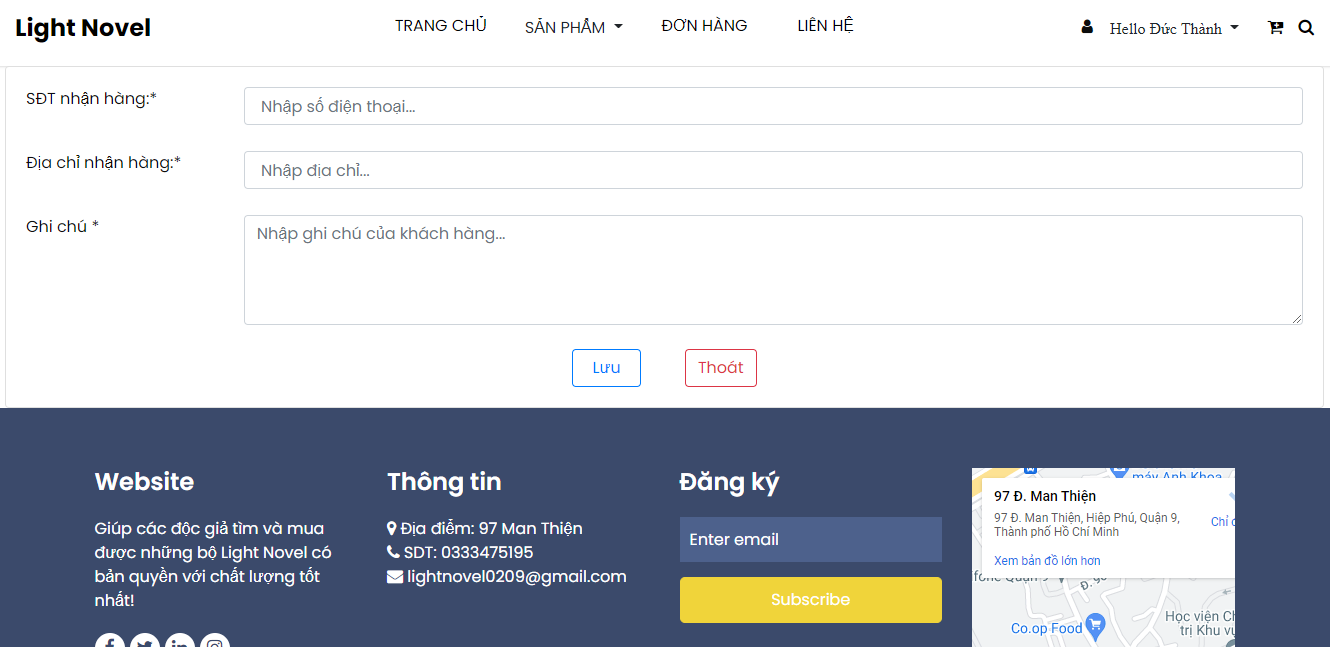


Hình 4.12 Màn hình giỏ hàng

* **Màn hình nhập thông tin nhận hàng:**

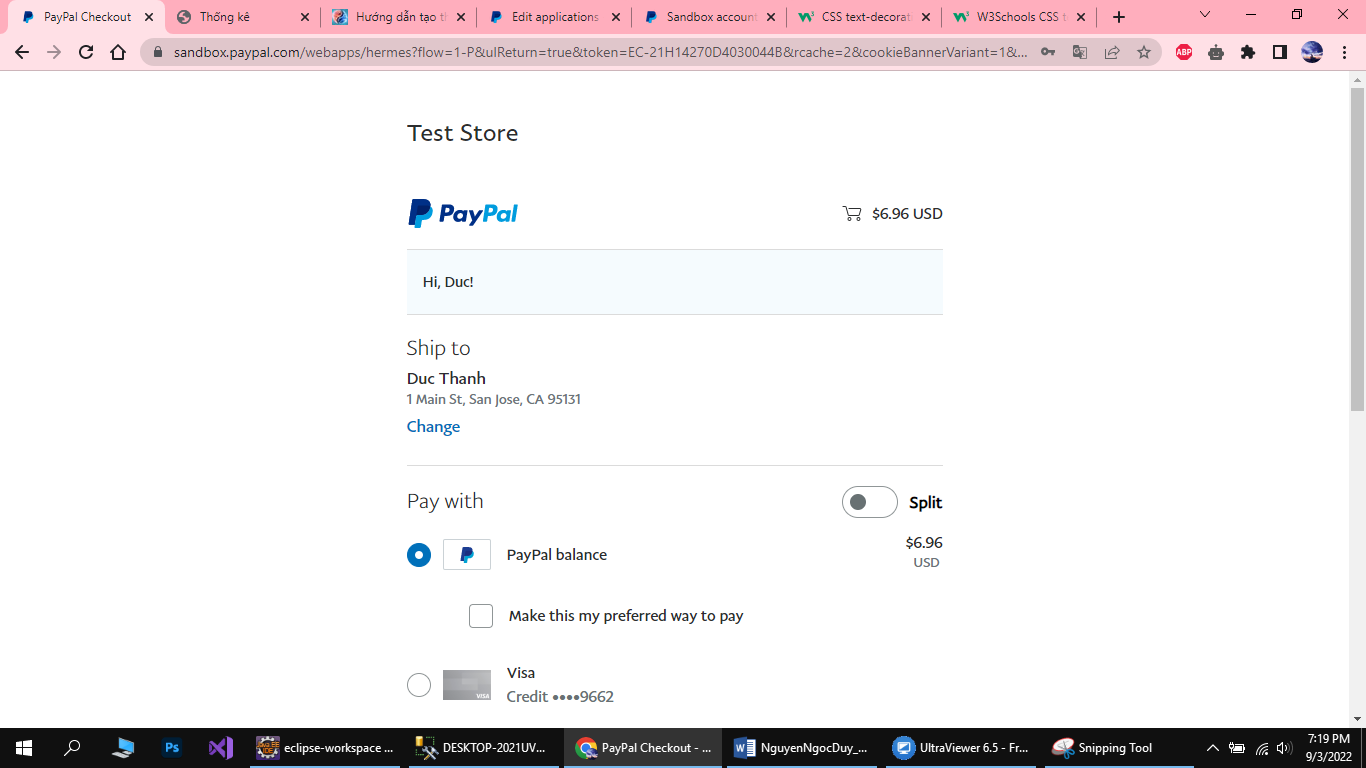
- Khách hàng nhập thông tin cá nhân để nhận hàng. Thông tin đơn hàng có:

* + - Số điện thoại nhận hàng
    - Địa chỉ nhận đơn hàng.
    - Ghi chú của khách hàng nếu muốn.



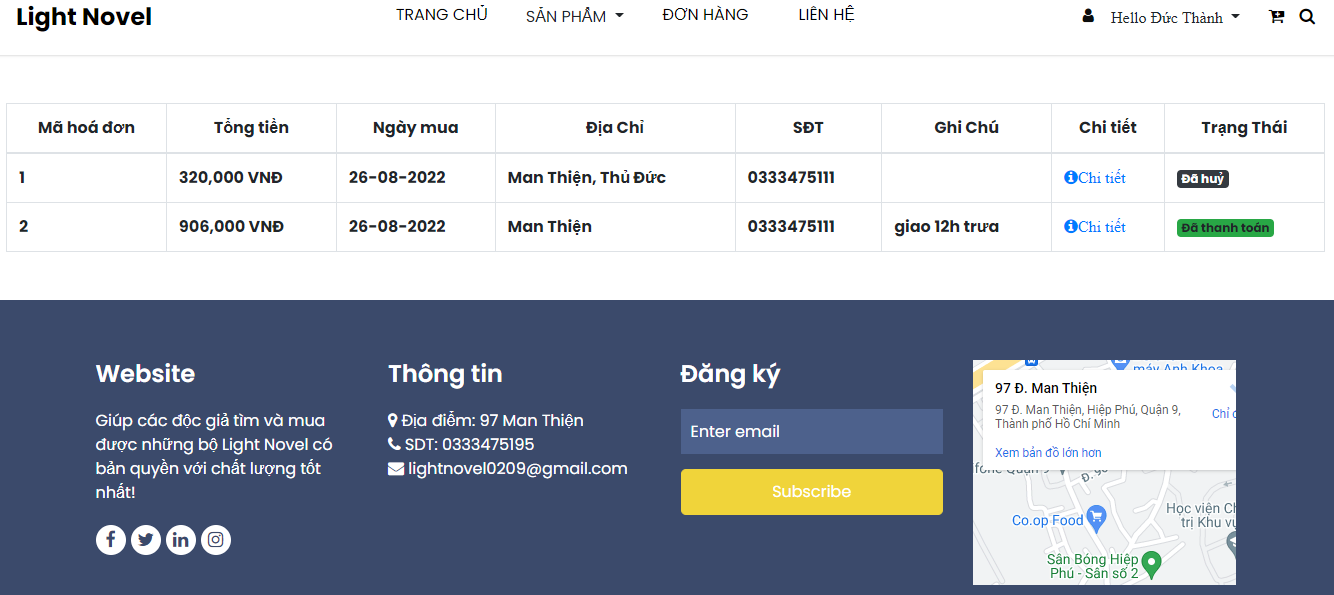
Hình 4.13 Màn hình nhập thông tin khách hàng để nhận hàng

* **Màn hình thanh toán online:**
* Khách hàng nhập login tài khoản paypal để tiến hành thanh toán.
  + - Đơn vị tiền tệ thanh toán là USD ( VND/23000)
    - Sau khi thanh toán sẽ báo thành công.



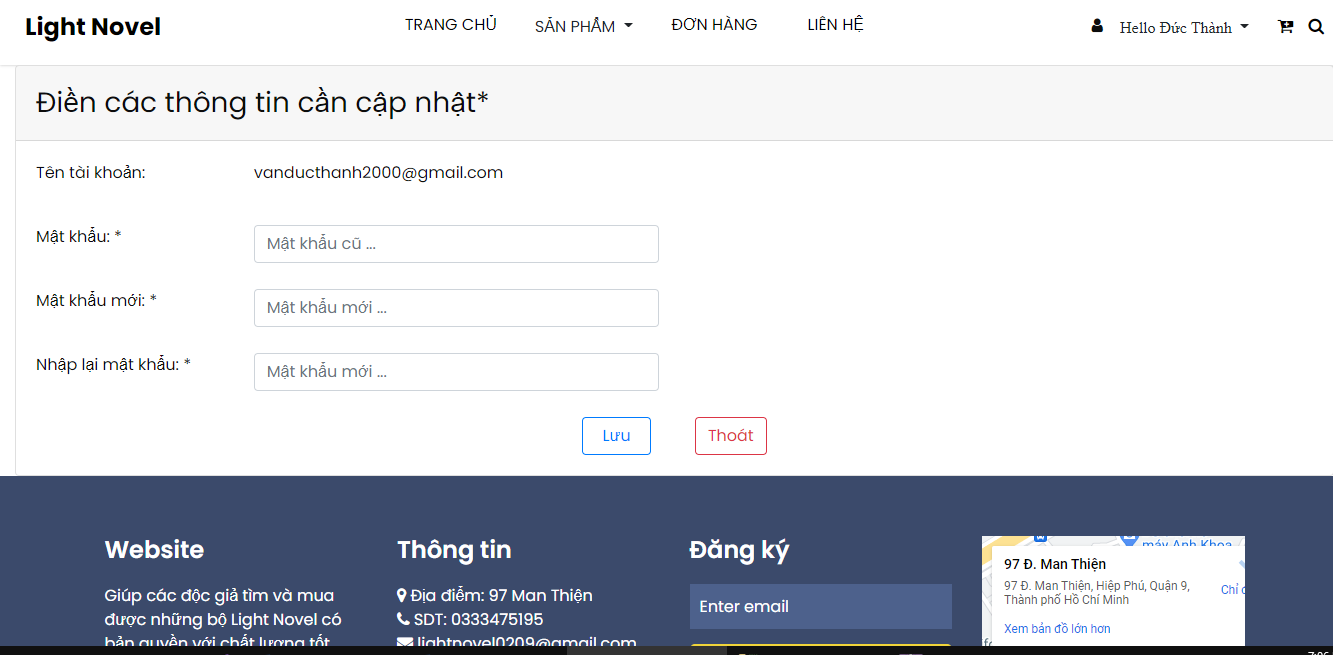
Hình 4.14 Màn hình thanh toán online

* **Màn hình chi tiết đơn hàng, lịch sử mua hàng:**
* Khách hàng có thể xem lịch sử mua hàng của mình, khi bấm vào từng đơn hàng sẽ xem được chi tiết đơn hàng.
* Thông tin đơn hàng có tổng tiền, ngày mua, địa chỉ và số điện thoai nhận hàng, ghi chú và trạng thái đơn hàng.
* Khách hàng có thể huỷ đơn hàng nếu admin chưa xác nhận đơn hàng.
* Thông tin chi tiết đơn hàng tương tự như thông tin ở giỏ hàng.



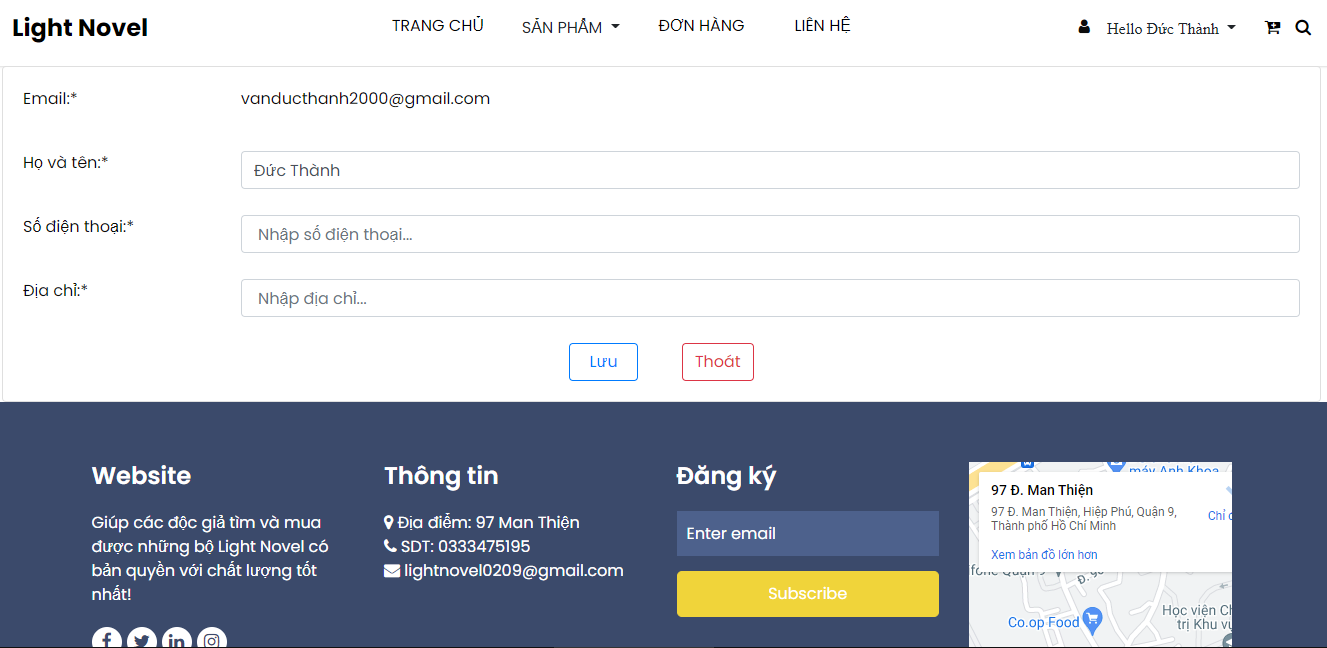
Hình 4.15 Màn hình lịch sử mua hàng

* **Màn hình thay đổi mật khẩu**
* Nhập mật khẩu cũ để hệ thống so sánh với dữ liệu trong database
* Nhập mật khẩu muốn đổi vào ô mật khẩu mới và xác nhân lại mật khẩu mới ở ô nhập lại mật khẩu



Hình 4.16 Màn hình đổi mật khẩu

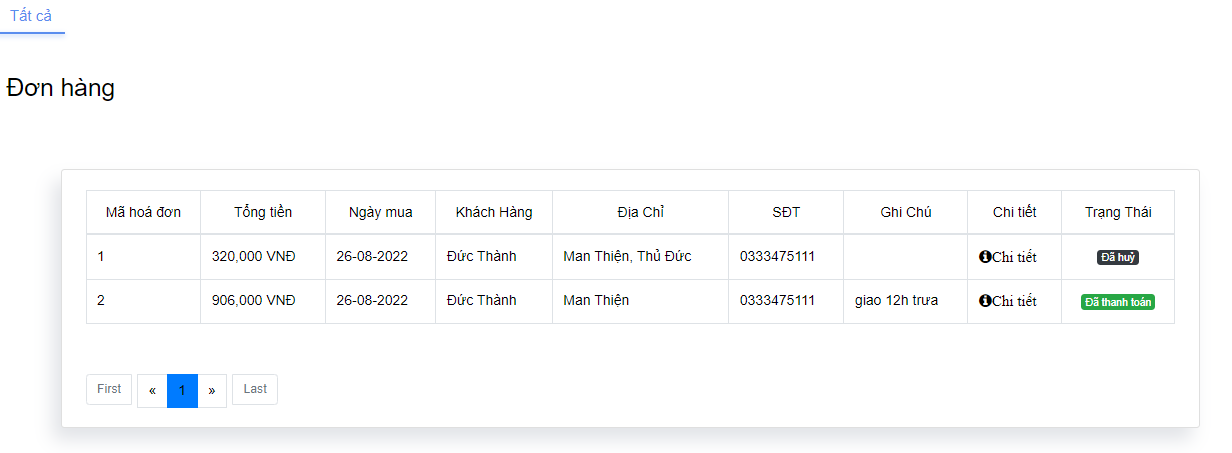
* **Màn hình thay đổi thông tin cá nhân**
* Hiển thị thông tin của khách hàng lên màn hình
* Nhập thông tin mới hoặc sửa thông tin cũ của khách hàng sau đó nhấn lưu để lưu lại thông tin.



Hình 4.17 Màn hình thay đổi thông tin cá nhân

### 4.2.2 Giao diện admin

* **Màn hình xử lý đơn hàng:**
* Đơn hàng sẽ được liệt kê theo từng trạng thái và theo tất cả đơn hàng. Thông tin về đơn hàng có:
* Mã đơn hàng
* Người nhận đơn hàng.
* Địa chỉ nhận hàng.
* Số điện thoại nhận hàng
* Ngày lập đơn hàng.
* Ghi chú
* Chi tiết đơn hàng.
* Trạng thái đơn hàng.
* Nhân viên có thể xác nhận đơn hàng hoặc huỷ đơn hàng.
* Khi nhân viên bấm xác nhận đơn hàng thì khách hàng không thể huỷ đơn.



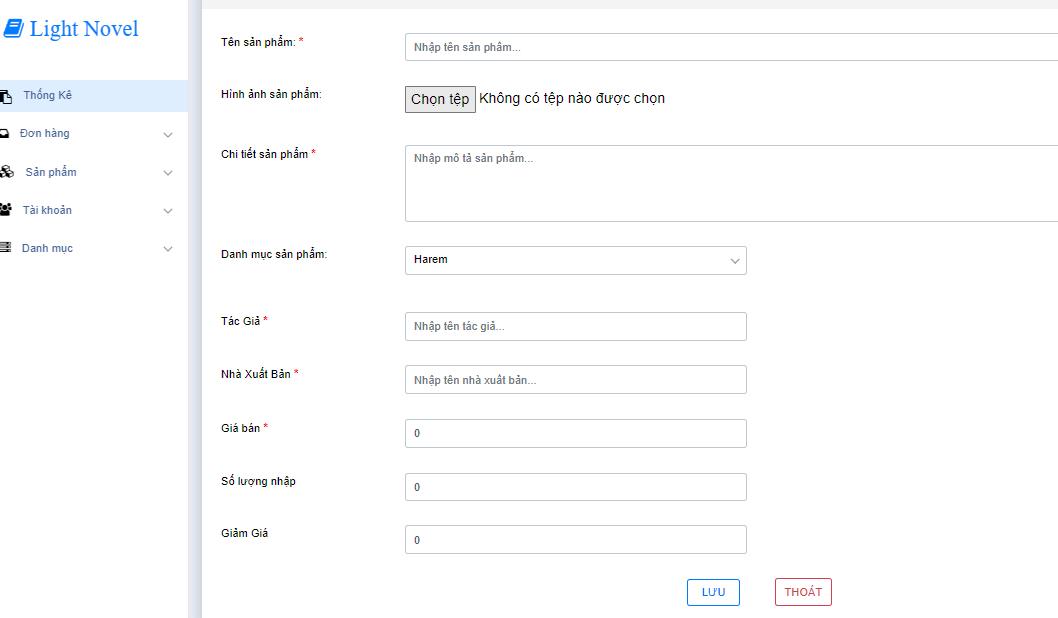
Hình 4.18 Màn hình quản lí đơn hàng

* **Màn hình quản trị sản phẩm:**
* Thông tin sản phẩm gồm:
* Tên sản phẩm.
* Loại sản phẩm.
* Đơn giá sản phẩm.
* Số lượng tồn
* Trạng thái sản phẩm
* Chức năng quản trị sản phẩm có thêm, xóa và cập nhật.



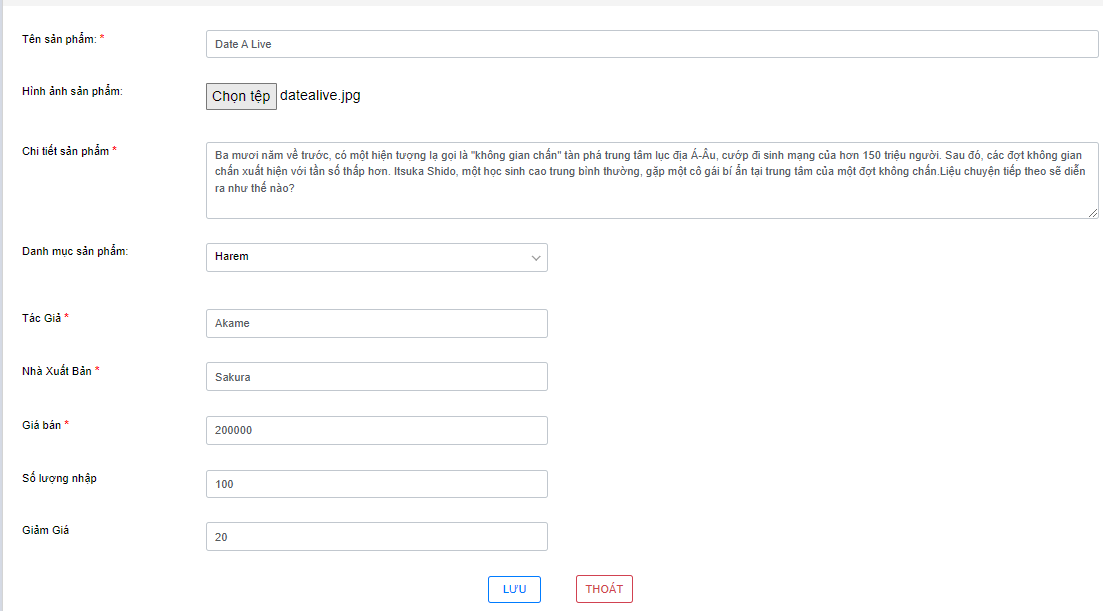
Hình 4.19 Màn hình quản lí sản phẩm

* **Màn hình thêm sản phẩm:**



Hình 4.20 Màn hình thêm sản phẩm

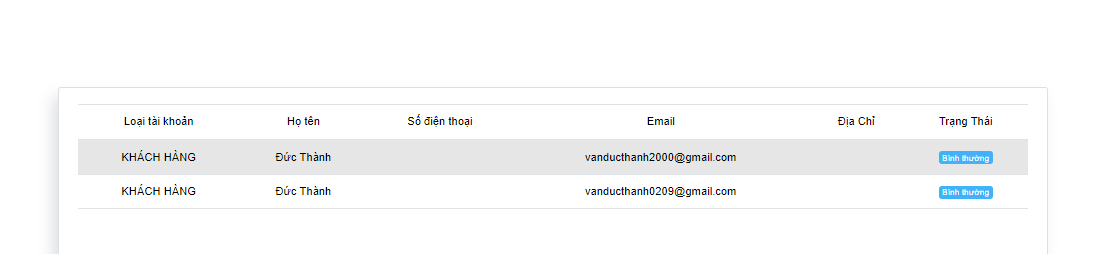
* **Màn hình sửa sản phẩm:**



Hình 4.21 Màn hình sửa sản phẩm

* **Màn hình danh sách khách hàng :**

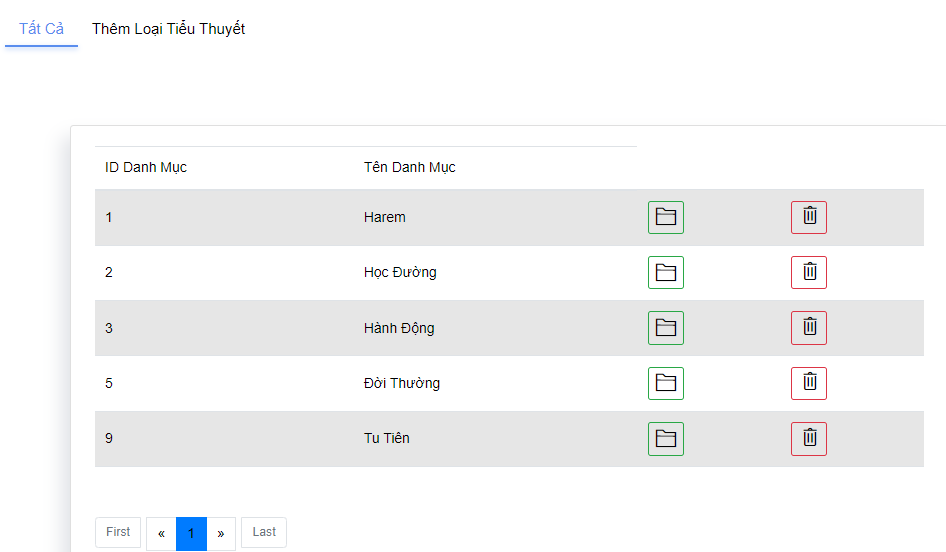
**-** In ra màn hình danh sách khách hàng của của hàng



Hình 4.22 Màn hình danh sách khách hàng

* **Màn hình quản trị danh mục sản phẩm:**
* Danh mục sản phẩm gồm:
* Mã danh mục .
* Tên danh mục.

- Chức năng quản trị danh mục có thêm xoá và cập nhật.



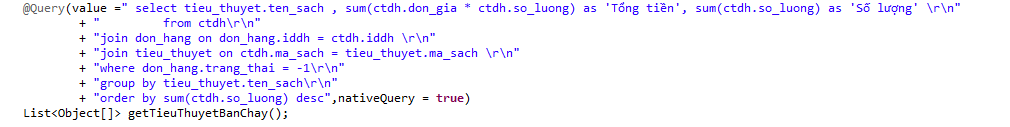
Hình 4.23 Màn hình quản trị danh mục sản phẩm

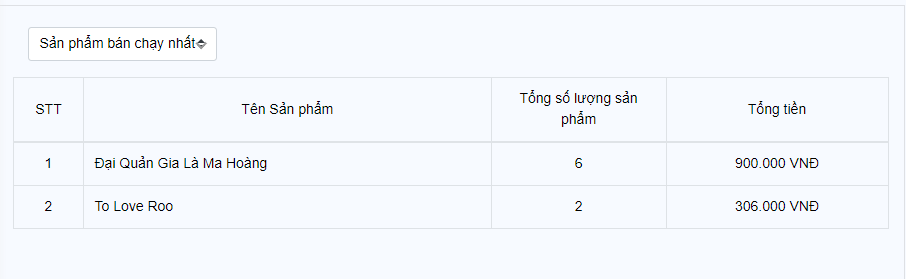
**Màn hình thống kê sản phẩm bán được:**

* Bao gồm:
* Tên sản phẩm .
* Số lượng bán được.

- Tổng tiền.

Câu query để thao tác với database để thống kê sản phẩm bán được





Hình 4.24 Màn hình thống kê sản phẩm bán chạy

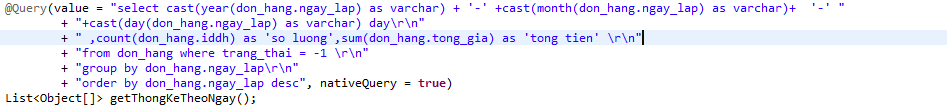
**Màn hình thống kê doanh thu theo ngày :**

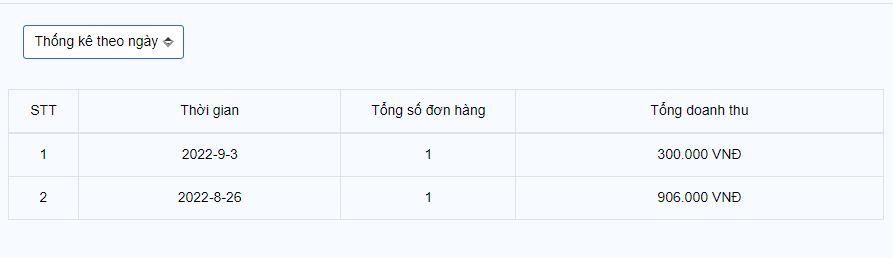
* Bao gồm:
* STT .
* Thời gian (năm/ tháng/ ngày).

- Tổng số đơn hàng.

- Tổng doanh thu.

Câu query để thao tác với database để thống kê doanh thu theo ngày





Hình 4.25 Màn hình thống kê doanh thu theo ngày

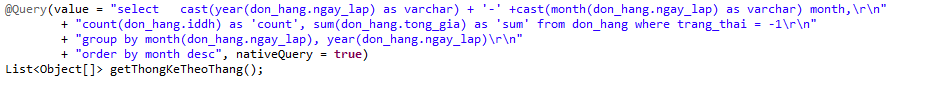
**Màn hình thống kê doanh thu theo tháng :**

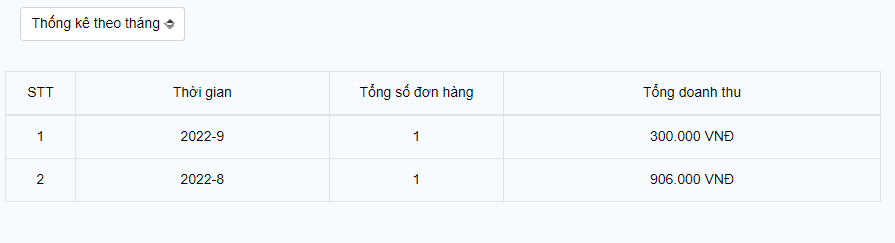
* Bao gồm:
* STT .
* Thời gian (năm/ tháng).

- Tổng số đơn hàng.

- Tổng doanh thu.

Câu query để thao tác với database để thống kê doanh thu theo tháng





Hình 4.26 Màn hình thống kê doanh thu theo tháng

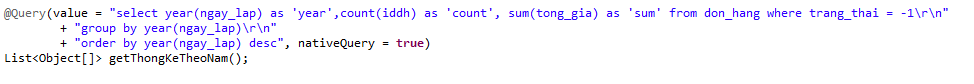
**Màn hình thống kê doanh thu theo năm:**

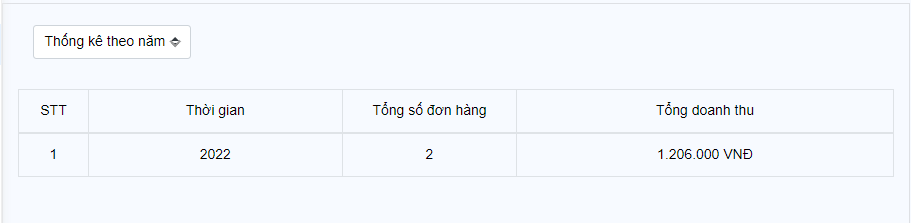
* Bao gồm:
* STT .
* Thời gian (năm).

- Tổng số đơn hàng.

- Tổng doanh thu.

Câu query để thao tác với database để thống kê doanh thu theo năm





Hình 4.27 Màn hình thống kê doanh thu theo năm

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong thời gian thực hiện đề tài dù gặp nhiều khó khăn nhưng em đã hoàn thành đề tài đúng thời gian quy định.

* **Kết quả đạt được:** 
  + - Xây dựng thành công API bằng Microsoft SQL Server và Spring Boot ..
    - Xây dựng thành công Website kinh doanh tiểu thuyết, đáp ứng được yêu cầu cơ bản của người sử dụng.
    - Xây dựng thành công cơ bản website quản trị.
    - Tìm hiểu tương đối kỹ về lập trình Java với công nghệ Spring.
    - Giao diện được thiết kế đơn giản và dễ sử dụng.
* **Những khó khăn và hạn chế trong quá trình thực hiện đề tài:**
* Thời gian thực hiện dự án gấp nên phân phối thời gian chưa hợp lý.
* Một số nội dung, kiến thức liên quan đến nền tảng cơ sở dữ liệu do chưa được học tập nên việc thiết kế và vận hành cơ sở dữ liệu còn ở mức sơ khai.
* Website vẫn còn một số modul chưa được hoàn thiện.
* **Hướng phát triển của đề tài:** Để tiếp tục phát triển đề tài này và có thể áp dụng trong thực tế, em nhận thấy cần phải tiếp tục thực hiện một số công việc sau**:**
* Xử lý các lỗi chặt chẽ hơn trước khi đưa vào sử dụng.
* Linh hoạt hơn trong các sự kiện và yêu cầu của khách hàng.
* Bổ sung nhiều hơn các tính năng cần thiết cho website.

- Hoàn thiện các chức năng phù hợp với thực tế hoạt động kinh doanh.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Các website tham khảo:**

* 1. Lập trình web : <https://laptrinhjavaweb.com/huong-dan-restful-web-service-trong-spring-boot-phan-1-17>

2. Youtube : <https://www.youtube.com/>

3. Thymeleaf: <https://www.thymeleaf.org/>

4. Học spring boot: [https://hocspringboot.net/](https://hocspringboot.net/%20)