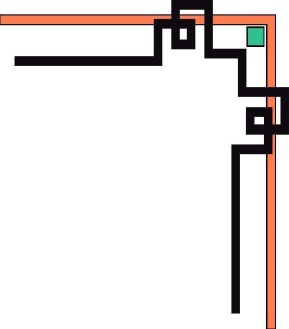
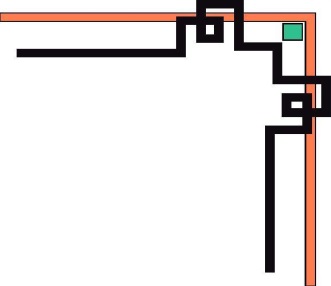
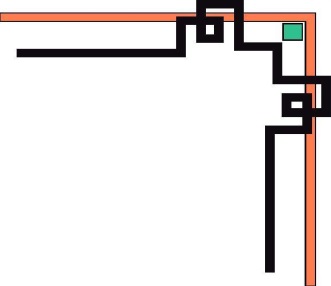
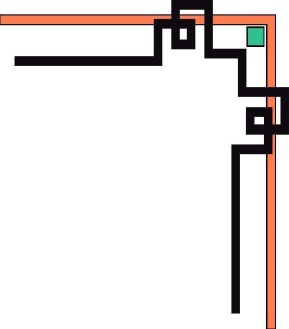
BỘ GIÁO DỤC VÀ ĐÀO TẠO



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**🕯✡🕮🕮✡🕯**



ĐỒ ÁN CÔNG NGHỆ THÔNG TIN

**XÂY DỰNG WEBSITE BÁN SÁCH**

**SỬ DỤNG FRAMEWORK DJANGO**

**GVHD**: **Trần Công Tú**

**SVTH: Hà Nhật Vềnh 20110599**

**Mã lớp học**: PROJ215879\_22\_1\_05CLC

Thành phố Hồ Chí Minh, Tháng 12 năm 2022

|  |  |
| --- | --- |
| **TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**  \*\*\* | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc** |

**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

**NHẬN XÉT CỦA GIÁO VIÊN:**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………

TP. Hồ Chí Minh, ngày …, tháng 12, năm 2022

Giáo viên chấm điểm

# MỤC LỤC

[MỤC LỤC 2](#_Toc122552130)

[DANH MỤC HÌNH ẢNH 3](#_Toc122552131)

[DANH MỤC BẢNG BIỂU 3](#_Toc122552132)

[DANH MỤC CÁC TỪ VIẾT TẮT 3](#_Toc122552133)

[NỘI DUNG 4](#_Toc122552134)

[Chương 1: Giới thiệu 4](#_Toc122552135)

[1.1. Lý do chọn đề tài: 4](#_Toc122552136)

[2.2. Khảo sát thị trường 5](#_Toc122552137)

[2.3. Tìm hiểu các website bán hàng [2] 6](#_Toc122552138)

[Chương 2: Công việc cần làm 7](#_Toc122552139)

[Chương 3: Cơ sở lý thuyết 9](#_Toc122552140)

[3.1. Cơ sở dữ liệu quan hệ là gì?[3] 9](#_Toc122552141)

[3.2. MySQL là gì? [4] 9](#_Toc122552142)

[3.3. Python và Django 10](#_Toc122552143)

[3.4. HTML, CSS, JavaScript là gì? [7] 13](#_Toc122552144)

[Chương 4: Thiết kế 15](#_Toc122552145)

[4.1. Thiết kế CSDL 15](#_Toc122552146)

[4.2. Thiết kế giao diện 18](#_Toc122552147)

[Chương 5: Cài đặt và Kiểm thử 25](#_Toc122552148)

[5.1. Cài đặt models trong django project để ánh xạ đến CSDL 25](#_Toc122552149)

[5.2. Cài đặt giao diện 29](#_Toc122552150)

[5.3. Các hàm xử lý chính 40](#_Toc122552151)

[KẾT LUẬN 42](#_Toc122552152)

[1. Kết quả đạt được sau khi thực hiện đề tài: 42](#_Toc122552153)

[2. Ưu nhược điểm của website 43](#_Toc122552154)

[Ưu điểm: 43](#_Toc122552155)

[Nhược điểm: 43](#_Toc122552156)

[3. Các chức năng có thể phát triển 43](#_Toc122552157)

[TÀI LIỆU THAM KHẢO 44](#_Toc122552158)

# DANH MỤC HÌNH ẢNH

[Hình 1: Thanh điều hướng 29](#_Toc122552403)

[Hình 2:Giao diện Trang chủ 33](#_Toc122552404)

[Hình 3: Giao diện Giỏ hàng 35](#_Toc122552405)

[Hình 4: Giao diện đặt hàng 37](#_Toc122552406)

[Hình 5: Giao diện chỉnh sửa thông tin 39](#_Toc122552407)

# DANH MỤC BẢNG BIỂU

[Bảng 1: [Công việc]Công việc cần làm 7](#_Toc122552083)

[Bảng 2: [Công việc]Các chức năng ở mỗi giao diện 8](#_Toc122552084)

[Bảng 3: Một số lệnh cơ bản trong manage.py 13](#_Toc122552085)

[Bảng 4: Bảng mô tả các thực thể 15](#_Toc122552086)

[Bảng 5: Một số phương thức xử lí 17](#_Toc122552087)

[Bảng 6: Bảng thiết kế giao diện 19](#_Toc122552088)

# DANH MỤC CÁC TỪ VIẾT TẮT

* KH: khách hàng
* TK: tài khoản
* SP: sản phẩm
* tt: thông tin

# NỘI DUNG

## Chương 1: Giới thiệu

### 1.1. Lý do chọn đề tài:

Ông Nguyễn Ngọc Dũng, Chủ tịch Hiệp hội TMĐT Việt Nam (Vecom) nhận xét, Covid-19 như một cú hích đẩy nhanh quá trình tăng trưởng của thương mại điện tử: “Từ khi dịch bùng phát, **nhu cầu mua sắm qua sàn thương mại điện tử tăng mạnh**. Đến nay, đã có hơn 70% dân số Việt Nam tiếp cận mạng Internet, trong đó, gần 50% người dùng Việt Nam đã mua sắm online,....” (theo baodautu.vn).[1]

Đây là lý do em chọn đề tài “*xây dựng website bán sách online”*

Đáp ứng được những nhu cầu mua sắm, em sẽ thiết kế một website thương mại điện tử nhằm phục vụ việc mua sắm online. Và để thiết kế một website nhanh chóng và hiệu quả thì sử dụng Framework Django là một trong những lựa chọn hàng đầu.

### 2.2. Khảo sát thị trường

Một số khảo sát về sở thích đọc sách:

Chart, pie chart

Description automatically generated

Khảo sát : Độ tuổi trả lời khảo sát

Timeline

Description automatically generated

Khảo sát : Thể loại (khảo sát)

Chart, bar chart

Description automatically generated

Khảo sát : Hình thức thanh toán (khảo sát)

### 2.3. Tìm hiểu các website bán hàng [2]

Qua quá trình tìm hiểu một số website bán hàng online, em tổng kết được những chức năng quan trọng mà một website bán hàng cần có:

* Thiết kế giao diện đẹp, dễ nhìn được các sản phẩm (thích hợp với điện thoại)
* Công cụ tiện ích (như là lọc, tìm kiếm sản phẩm)
* Hình thức thanh toán đa dạng như: Thanh toán trực tuyến, thanh toán khi nhận hàng, được kiểm tra hàng trước khi thanh toán,…
* Liên kết, quảng cáo bán hàng đến các mạng xã hội (facebook, google,…)
* Tính năng cho khách hàng (xem đơn hàng, đánh giá sản phẩm)

## Chương 2: Công việc cần làm

Từ những [tìm hiểu về website bán hàng](#_2.3._Tìm_hiểu) và một số khảo sát cho thấy để thiết kết được website có các chức năng cần thiết phải làm những công việc cụ thể sau:

Bảng 1: [Công việc]Công việc cần làm

|  |  |  |
| --- | --- | --- |
| **Công việc** | Chi tiết/ mục đích | Hoàn thành |
| Tìm hiểu cơ sở lý thuyết | - Ngôn ngữ lập trình: python, HTML, CSS, JavaScript.  - Framework Django  🡪 Mục đích: đây là cơ sở lý thuyết cần có để có thể thực hiện thiết kế website | 100% |
| Thiết kế CSDL | Các thực thể:  - Người dùng (User): khách hàng, quản trị (admin)  - Sản phẩm (Product\_book)  - Tác giả (Author)  - Hóa đơn (Invoice): *trong hóa đơn sẽ có nhiều order*  - Order: *chứa số lượng đặt của một khách hàng*  (được sử dụng như order Item)  🡪 Nhằm lưu trữ các dữ liệu của website | 100% |
| Thiết kế giao diện, chức năng | Gồm các giao diện chính:   * Trang chủ (show sản phẩm, thanh điều hướng) * Giỏ hàng (chứa sản phẩm khách hàng đã đặt) * Đặt hàng (tổng quan lại đơn hàng để KH thanh toán)   Một số giao diện khác:   * Tài khoản (cho phép khách sửa thông tin TK) * Xem đơn hàng (liệt kê các thông tin đơn hàng mà khách hàng đã đặt trước đó) * Giao diện đăng nhập, đăng ký TK   🡪 Giao diện để người dùng tương tác với website | 100% |

Bảng 2: [Công việc]Các chức năng ở mỗi giao diện

|  |  |  |
| --- | --- | --- |
| **Giao diện** | **Chức năng** | **Hoàn thành** |
| Trang chủ | Thanh điều hướng (navigation bar) | 100% |
| Hiện thị tt sản phẩm (cả lúc KH chưa đăng nhập) | 100% |
| Lọc sản phẩm (theo danh mục, theo từ khóa) | 100% |
| Thêm nhanh sản phẩm vào giỏ hàng | 100% |
| Xem chi tiết một sản phẩm (tác giả, đánh giá, thêm một *lượng* sản phẩm vào giỏ hàng) | 100% |
| Giỏ hàng | Liệt kê các sản phẩm đã được thêm vào giỏ | 100% |
| Có nút cập nhật (tăng, giảm, xóa) số lượng | 100% |
| Xem tổng số lượng, tổng giá | 100% |
| Thanh toán | Liệt kê lại các sản phẩm đã order | 100% |
| Cho phép nhập địa chỉ nhận hàng (mặc định địa chỉ của KH) | 100% |
| Tài khoản | Chỉnh sửa lại thông tin khách hàng | 100% |
| Xem đơn hàng | Liệt kê tt các đơn hàng khách đã đặt | 100% |
| Có thể để lại bình luận ở mỗi sản phẩm đã mua | 100% |
| Đăng nhập | Có thể đăng nhập bằng tài khoản Google | 100% |
| Đăng ký | Đăng ký bằng cách nhập thông tin hợp lệ |  |
| \_\_ | Hiện thị thông báo cho các tác vụ người dùng | 100% |

## Chương 3: Cơ sở lý thuyết

### 3.1. Cơ sở dữ liệu quan hệ là gì?[3]

Cơ sở dữ liệu quan hệ (tiếng Anh: relational database) là một cơ sở dữ liệu (phổ biến nhất là kỹ thuật số) dựa trên mô hình quan hệ dữ liệu, theo đề xuất của Edgar F. Codd vào năm 1970. Một hệ thống phần mềm sử dụng để duy trì cơ sở dữ liệu quan hệ là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Nhiều hệ thống cơ sở dữ liệu quan hệ có tùy chọn sử dụng SQL (Ngôn ngữ truy vấn có cấu trúc) tiêu chuẩn để truy vấn và duy trì cơ sở dữ liệu.

### 3.2. MySQL là gì? [4]

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

#### 3.2.1. Mô hình client-server

Diagram

Description automatically generated with low confidence

Một hoặc nhiều thiết bị (máy khách) kết nối với máy chủ thông qua một mạng cụ thể. Mọi máy khách có thể đưa ra yêu cầu từ giao diện người dùng trên màn hình của họ và máy chủ sẽ tạo ra kết quả mong muốn, miễn là cả hai đầu đều hiểu hướng dẫn. Các quy trình chính diễn ra trong môi trường MySQL đều giống nhau, đó là:

* MySQL tạo cơ sở dữ liệu để lưu trữ và thao tác dữ liệu, xác định mối quan hệ của từng bảng.
* Client có thể đưa ra yêu cầu bằng cách nhập các câu lệnh SQL cụ thể trên MySQL.
* Server application sẽ phản hồi với thông tin được yêu cầu và nó sẽ xuất hiện ở phía máy khách.

Ở phía Client, sẽ để ý đến MySQL GUI (Giao diện đồ họa người dùng) sử dụng. GUI càng nhẹ và thân thiện với người dùng, các hoạt động quản lý dữ liệu của họ sẽ nhanh hơn và dễ dàng hơn. Một số MySQL GUI phổ biến nhất là , SequelPro, DBVisualizer và Công cụ quản trị Navicat DB…

#### 3.2.2. Ưu điểm và nhược điểm của MySQL

##### *6.2.1. Ưu điểm*

* An toàn: Vì MySQL sở hữu nhiều tính năng bảo mật cấp cao, mã hóa thông tin đăng nhập và chứng thực từ host đều khả dụng.
* Dễ sử dụng: MySQL ổn định và dễ sử dụng trên nhiều hệ điều hành và cung cấp một hệ thống các hàm tiện ích lớn.
* Khả năng mở rộng: Với MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa có thể mở rộng khi cần thiết.
* Hiệu năng cao: Hỗ trợ nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS).

##### *6.2.2. Nhược điểm*

* Giới hạn: Nó vẫn bị hạn chế về một số chức năng cần thiết.
* Dung lượng hạn chế: Nếu só bản khi càng lớn thì việc truy xuất dữ liệu sẽ phức tạp và khó khăn hơn. Bạn cần phải áp dụng nhiều thủ thuật để nâng cấp tốc độ truy xuất dữ liệu lên.

### 3.3. Python và Django

#### 3.3.1 Python là gì?

Logo, company name

Description automatically generatedPython là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

#### 3.3.2. Những lợi ích của Python [5]

* Các nhà phát triển có thể dễ dàng đọc và hiểu một phần trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
* Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một phần trình Python.
* Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
* Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
* Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
* Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
* Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

#### 3.3.3. Django là gì? [6]

Django là 1 web framework khá nổi tiếng được viết hoàn toàn bằng ngôn ngữ Python. Nó là 1 framework với đầu đủ các thư viện, module hỗ trợ các web-developer. Django sử dụng mô hình MVC và được phát triển bởi Django Software Foundation (DSF một tổ chức phi lợi nhuận độc lập) Mục tiêu chính của Django là đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website phổ biến được xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket.

##### Giới thiệu mô hình MVC

Mô hình MVC đã được phát triển trong thời gian rất dài, và nó càng phát triển mạnh hơn nữa trong thời đại Internet, khi mà các ứng dụng Web ngày càng phổ biến. Internet chính là một nơi lý tưởng để triển khai mô hình client-server.

* Model (M) là mô phỏng của dữ liệu. Nó không thực sự là dữ liệu, nhưng nó là một thể hiện của dữ liệu và là nơi để chúng ta thao tác với dữ liệu thật sự.
* View (V) là những gì người dùng nhìn thấy. Nó là sự thể hiện của dữ liệu đối với người dùng. Nói một cách văn hoa, nó là sự thể hiện của Model.
* Controller (C) dùng để điều khiển luồng thông tin dữ Model và View. Nó được sử dụng để cài đặt các login về việc lấy dữ liệu từ DB thông qua Model và chuyển sang View. Nó cũng là nơi xử lý những truy vấn từ người dùng thông qua View và thực hiện các logic khác: thay đổi View, cập nhật dữ liệu thông qua Model.

##### Cấu trúc của một Django project

* Để tạo một project Django cần mở Command Prompt (cmd) lên, chuyển đến thư mục mà bạn muốn tạo, sau đó gõ đoạn lệnh django-andmin startproject <tên thư mục>. Khi đặt tên project tránh đặt những tên trùng với các từ khóa có sẵn trong Python như sys, os, django…. để tránh bị xung đột.
* Ví dụ tạo project tên testproj, thư mục được tạo ra có cấu trúc:

Graphical user interface, text, application, Word

Description automatically generated

* Trong đó:
* testproj/: ngoài cùng là root directory chứa project.
* manage.py: đây là 1 command-line utility cho phép mình thao tác với project theo nhiều cách hiện hành của mình.
* testproj/: bên trong là gói các package Python sẽ sử dụng cho project:
* testproj/\_\_init\_\_.py: là 1 file rỗng chỉ định việc cái đường dẫn này sẽ được xem như là 1 Python package.
* testproj/settings.py: Settings/configuration cho project Django
* testproj/urls.py: khai báo URL cho project
* testproj/wsgi.py: entry-point cho tích hợp web server WSGI

##### Một số lệnh trong manage.py

Bảng : Một số lệnh cơ bản trong manage.py

|  |  |  |
| --- | --- | --- |
| TT | Lệnh | Mục đích |
| 1 | django-admin startproject <Tên Project> | Tạo ra thư mục chứa django project |
| 2 | py manage.py startapp <app\_name> | Tạo ra thư mục, nơi để tạo ra những trang web nằm trong toàn bộ hệ thống website. |
| 3 | py manage.py runserver | Chạy server |
| 4 | py manage.py makemigrations <app\_name>  py manage.py migrate | Khi chỉnh sửa trong models và cần cập nhật lại CSDL. |

### 3.4. HTML, CSS, JavaScript là gì? [7]

**HTML** là viết tắt của cụm từ Hypertext Markup Language (tạm dịch là Ngôn ngữ đánh dấu siêu văn bản). HTML được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes… và HTML không phải là ngôn ngữ lập trình.

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

**JavaScript** là ngôn ngữ lập trình website phổ biến hiện nay, nó được tích hợp và nhúng vào HTML giúp website trở nên sống động hơn. JavaScript đóng vai trò như là một phần của trang web, thực thi cho phép Client-side script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động.

Đây là 3 ngôn ngữ chính trong lập trình web và có mối liên hệ lẫn nhau để xây dựng một website sống động, chuyên nghiệp:

* HTML: Hỗ trợ trong việc xây dựng layout, thêm nội dung dễ dàng trên website.
* CSS: Hỗ trợ việc định dạng thiết kế, bố cục, style, màu sắc,…
* JavaScript: Tạo nên những nội dung “động” trên website.

## Chương 4: Thiết kế

### 4.1. Thiết kế CSDL

#### 4.1.1. Mô hình ERD: Từ những chức năng chính

Diagram

Description automatically generated

Hình 1: Mô hình ERD website bán sách bằng framework django

#### 4.1.2. Mô tả các trường của CSDL

Bảng : Bảng mô tả các thực thể

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Thực thể | Các thuộc tính | Ghi chú |
| 1 | Product | **pID** | Mã sản phẩm, để xác định sản 1 sản phẩm duy nhất |
| Book\_name (varchar(100)) | Tên sản phẩm (sách) |
| catID – FK (int) | Khóa ngoại tham chiếu đến bảng Category |
| Book\_img (varchar) | Chứa vị trí hình của sản phẩm |
| Book\_price (int) | Giá của sản phẩm |
| Book\_description (longtext) | Mô tả của sản phẩm |
| 2 | Author | **auID (int)** | Mã tác giả, xác định duy nhất |
| Au\_name (varchar(50)) | Tên tác giả |
| 3 | Category | **catID (int)** | Mã loại sách |
| Cat\_name (varchar(50)) | Tên loại sách (tham khảo, giáo trình,…) |
| 4 | OrderItem | **oID (int)** | Mã Order |
| pID – FK (int) | Khóa ngoại tham chiếu đến product |
| iID – FK (int) | Khóa ngoại tham chiếu đến invoice |
| Quantity (smallint) | Số lượng đặt |
| Comment (varchar(254)) | Bình luận của khách hàng về sản phẩm |
| 5 | Invoice | **iID (int)** | Mã hóa đơn |
| cusID - FK | Khóa ngoại tham chiếu đến bảng khách hàng |
| Date (date) | Ngày hóa đơn được tạo ra |
| date\_checkout (date) | Ngày khách hàng bấm đặt hàng |
| Place\_status (boolean) | Kiểm tra xem đơn hàng đã đc đặt chưa (mặc định là False |
| Ship\_addr (varchar(150)) | Địa chỉ đặt hàng (mặc định theo địa chỉ KH) |

#### 4.1.3. Công thức tính toán trong models

Bảng : Một số phương thức xử lí

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Nội dung | Vị trí | Mục đích |
| 1 | def get\_total(*self*):          total = *self*.pID.book\_price \* *self*.quantity          return total | store/models.py  dòng 132 | Tính tổng giá của một order.  Vd: order 2 cuốn sách 50k  Kết quả trả về: 100k |
| 2 | def get\_total\_item(*self*):          orders = *self*.order\_set.all()          total = sum([i.quantity for i in orders])          return total | store/models.py  dòng 109 | Tính tổng số lượng sản phẩm của đơn hàng  Vd: 2 cuốn A, 1 cuốn B  Trả về: 3 |
| 3 | def get\_total\_price(*self*):          orders = *self*.order\_set.all()          total = sum([i.get\_total for i in orders])          return total | store/models.py  dòng 115 | Tính tổng giá tiền của đơn hàng (tính tổng lại của hàm **get\_total**)  Vd: 2 cuốn A 50k, 1 cuốn B 20k  Trả về: 120k |

#### 4.1.4. Sơ đồ luồng dữ liệu (data flow diagram - DFD)

Diagram

Description automatically generated

### 4.2. Thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Giao diện | Đường dẫn | Mục đích |
| 1 | Base.html |  | Để các template khác tái sử dụng (gồm có header, và block content sẽ đưa dữ liệu vào base)  Header sẽ có các đường dẫn:   * Home (logo) để quay về trang chủ * Tài khoản   + Đăng nhập   + Đăng xuất   + Xêm đơn hàng đã đặt * Thanh lọc sản phẩm * Giỏ hàng |
| 2 | Index.html  (Trang chủ) | / | Show các sản phẩm của cửa hàng   * Thêm sản phẩm vào giỏ hàng * Xem chi tiết sản phẩm |
| 3 | Cart.html  (Giỏ hàng) | /cart | Show các sản phẩm mà khách hàng đã thêm vào giỏ, có nút “Thanh toán” để chuyển sang giao diện đặt hàng. |
| 4 | Checkout.html  (Đặt hàng) | /checkout | Giao diện đặt hàng cho phép khách nhập địa chỉ nhận hàng và xem lại toàn bộ sản phẩm có trong giỏ hàng. |
| 5 | Login|Register | /login  /register | Giao diện đăng nhập gồm: username - pass  Giao diện đăng ký: Kế thừa lớp **forms** trong django để cho phép khách hàng đăng ký.  Có nút đăng nhập bằng google |

Bảng : Bảng thiết kế giao diện

#### 4.2.1. base.html

Vị trí: \store\templates\layouts\base.html

<!doctype html>

<html *lang*="vi">

{% load static %}

<head>

  {% block title %}<title>HOME PAGE</title>{% endblock %}

  <!-- Required meta tags -->

  <meta *charset*="utf-8">

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

  <!-- Bootstrap CSS v5.2.1 -->

  <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" *rel*="stylesheet"

*integrity*="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" *crossorigin*="anonymous">

  <script *src*="https://kit.fontawesome.com/f9029665cb.js" *crossorigin*="anonymous"></script>

  <link *rel*="stylesheet" *href*="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.2/font/bootstrap-icons.css">

  <link *rel*="stylesheet" *href*="{% static 'css/mystyle.css' %}">

</head>

<body *style*="padding-top: 120px">

  <!-- ========== Start Header ========== -->

  {% include 'includes/header.html' %}

  <!-- ========== End Header ========== -->

  {% include 'includes/message.html' %}

  <!-- ========== Start Main ========== -->

  {% block content %} {% endblock content %}

  <!-- ========== End Main ========== -->

  <footer>

  </footer>

  <!-- Bootstrap JavaScript Libraries -->

  <script *src*="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"

*integrity*="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" *crossorigin*="anonymous">

    </script>

  <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"

*integrity*="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz" *crossorigin*="anonymous">

    </script>

  <script *type*="text/javascript" *src*="{% static 'js/cart.js' %}"></script>

  <script *type*="text/javascript">

    var user = '{{request.user}}'

    function getToken(*name*) {

      let cookieValue = null;

      if (document.cookie && document.cookie !== '') {

        const cookies = document.cookie.split(';');

        for (let i = 0; i < cookies.length; i++) {

          const cookie = cookies[i].trim();

          // Does this cookie string begin with the name we want?

          if (cookie.substring(0, name.length + 1) === (name + '=')) {

            cookieValue = decodeURIComponent(cookie.substring(name.length + 1));

            break;

          }

        }

      }

      return cookieValue;

    }

    const csrftoken = getToken('csrftoken');

  </script>

</body>

</html>

#### 4.2.2. cart.html

{% extends 'layouts/base.html' %} {% load static %} {% load humanize %}

{% block content %}

<div *class*="row">

  <div *class*="col-lg-12">

    <div *class*="box-element">

      <a *class*="btn btn-outline-dark" *href*="{% url 'store:home' %}">&#x2190;Quay lại</a>

      <br />

      <br />

      <table *class*="table">

        <tr>

          <th>

            <h5>Số lượng: <strong>{{invoice.get\_total\_item}}</strong></h5>

          </th>

          <th>

            <h5>Tổng cộng: <strong>{{invoice.get\_total\_price|intcomma}}đ</strong></h5>

          </th>

          <th>

            <a *style*="float: right; margin: 5px" *class*="btn btn-success" *href*="{% url 'store:checkout' %}" >Đặt hàng</a>

          </th>

        </tr>

      </table>

    </div>

    <br />

    <div *class*="box-element">

      <div *class*="cart-row">

        <div *style*="flex: 2"></div>

        <div *style*="flex: 2"><strong>Sản phẩm</strong></div>

        <div *style*="flex: 1"><strong>Giá</strong></div>

        <div *style*="flex: 1"><strong>Số lượng</strong></div>

        <div *style*="flex: 1"><strong>Tổng cộng</strong></div>

        <div *style*="flex: 1"><strong>Xóa</strong></div>

      </div>

      {% for i in orders %}

      <div *class*="cart-row">

        <div *style*="flex: 2">

          <img *class*="row-image" *src*="{{i.pID.book\_img.url}}" />

        </div>

        <div *style*="flex: 2"><p>{{i.pID.book\_name}}</p></div>

        <div *style*="flex: 1"><p>{{i.pID.book\_price}}</p></div>

        <div *style*="flex: 1">

          <p *class*="quantity">{{i.quantity}}</p>

          <div *class*="quantity">

            <img *data-product*="{{i.pID.pID}}" *data-action*="add" *class*="chg-quantity update-cart" *src*="{% static 'img/arrow-up.png' %}"/>

            <img *data-product*="{{i.pID.pID}}" *data-action*="remove" *class*="chg-quantity update-cart" *src*="{% static 'img/arrow-down.png' %}"/>

          </div>

        </div>

        <div *style*="flex: 1"><p>{{i.get\_total}}</p></div>

        <div *style*="flex: 1">

          <a *data-product*="{{i.pID.pID}}" *data-action*="delete" *class*="chg-quantity update-cart btn-close" *type*="button" ></a>

        </div>

      </div>

      {% endfor %}

    </div>

  </div>

</div>

{% endblock content %}

#### 4.2.3. checkout.html

{% extends 'layouts/base.html' %} {% load static %} {% load humanize %}

{% block content %}

<div *class*="row">

  <a *class*="btn btn-outline-dark" *href*="{% url 'store:cart' %}"

    >&#x2190; Quay về giỏ hàng</a

  >

  {% if user.username %} {% if orders %}

  <div *class*="col-lg-6">

    <div *class*="container box-element">

      <form *action*="{% url 'store:checkout\_submit' %}" *method*="POST">

        {% csrf\_token %}

        <h2 *class*="text-center">Thông tin giao nhận</h2>

        <div *class*="mb-3 mt-3">

          <label *for*="ship\_addr" *class*="form-label">Địa chỉ nhận hàng:</label>

          <input

*id*="ship\_addr"

*name*="ship\_addr"

*value*="{{invoice.cusID.cus\_addr}}"

*type*="text"

*class*="form-control"

*placeholder*="Nhập địa chỉ nhận hàng"

          />

        </div>

        <input *type*="hidden" *name*="iID" *value*="{{invoice.iID}}" />

        <button *type*="submit" *class*="btn btn-primary">Đặt hàng</button>

      </form>

    </div>

  </div>

  {% else %}

  <div *class*="col-lg-6">

    <div *class*="alert alert-danger" *role*="alert">

      Giỏ hàng của bạn đang trống

      <a *href*="{% url 'store:home' %}" *class*="alert-link">quay lại</a> để mua

      hàng!

    </div>

  </div>

  {% endif %}

  <div *class*="col-lg-6">

    <div *class*="box-element">

      <p>Số lượng sản phẩm: <strong>{{invoice.get\_total\_item}}</strong></p>

      <p>Tổng cộng: <strong>{{invoice.get\_total\_price|intcomma}}đ</strong></p>

      <div *class*="cart-row">

        <div *style*="flex: 2"><img *class*="row-image" *src*="" /></div>

        <div *style*="flex: 2">

          <p>Tên sách</p>

        </div>

        <div *style*="flex: 1">

          <p>Giá</p>

        </div>

        <div *style*="flex: 1">

          <p>Số lượng</p>

        </div>

      </div>

      {% for i in orders %}

      <div *class*="cart-row">

        <div *style*="flex: 2">

          <img *class*="row-image" *src*="{{ i.pID.book\_img.url}}" />

        </div>

        <div *style*="flex: 2">

          <p>{{i.pID.book\_name}}</p>

        </div>

        <div *style*="flex: 1">

          <p>{{i.pID.book\_price|intcomma}}đ</p>

        </div>

        <div *style*="flex: 1">

          <p>{{i.quantity}}</p>

        </div>

      </div>

      {% endfor %}

    </div>

  </div>

  <!-- ========== Start Thông báo cho khách nếu chưa đăng nhập ========== -->

  {% else %}

  <div *class*="alert alert-danger" *role*="alert">

    Bạn cần

    <a *href*="{% url 'store:login' %}" *class*="alert-link">đăng nhập</a> hoặc

    <a *href*="{% url 'store:register' %}" *class*="alert-link">đăng ký</a> để mua

    hàng!

  </div>

  {% endif %}

  <!-- ========== End Thông báo cho khách nếu chưa đăng nhập ========== -->

</div>

{% endblock content %}

#### 4.2.4. edit\_profile.html

{% extends 'layouts/base.html' %}

{% block title %}<title>EDIT ACCOUNT</title>{% endblock %}

{% block content %}

    <div *class*="container" *style*="width:500px">

        <form *action*="{% url 'store:edit\_profile' %}" *method*="post">

            {% csrf\_token %}

            <div *class*="mb-3">

                <label *for*="cus\_name" *class*="form-label">username</label>

                <input *disabled* *name*="cus\_name" *type*="text" *class*="form-control" *id*="cus\_name" *placeholder*="Họ tên" *value*="{{user.username}}">

                <label *for*="user\_email" *class*="form-label">Email</label>

                <input *disabled* *name*="user\_email" *type*="text" *class*="form-control" *id*="user\_email" *placeholder*="Nhập email của bạn" *value*="{{user.email}}">

            </div>

            <div *class*="mb-3">

              <label *for*="cus\_name" *class*="form-label">Họ tên</label>

              <input *name*="cus\_name" *type*="text" *class*="form-control" *id*="cus\_name" *placeholder*="Họ tên" *value*="{{cus.cus\_name}}">

            </div>

            <div *class*="mb-3">

                <label *for*="cus\_addr" *class*="form-label">Địa chỉ</label>

                <input *name*="cus\_addr" *type*="text" *class*="form-control" *id*="cus\_addr" *placeholder*="Địa chỉ" *value*="{{cus.cus\_addr}}">

            </div>

            <div *class*="mb-3">

                <label *for*="cus\_phone" *class*="form-label">Số điện thoại</label>

                <input *name*="cus\_phone" *type*="text" *class*="form-control" *id*="cus\_phone" *placeholder*="Số điện thoại" *value*="{{cus.cus\_phone}}">

            </div>

            <button *type*="submit" *class*="btn btn-primary rounded">Sửa thông tin</button>

          </form>

    </div>

{% endblock content %}

## Chương 5: Cài đặt và Kiểm thử

### 5.1. Cài đặt models trong django project để ánh xạ đến CSDL

*Vị trí: store\models.py*

from django.db import models

from django.contrib.auth.models import User

from django.urls import reverse

class Category(*models*.Model):

    catID = models.AutoField(*primary\_key*=True)

    cat\_name = models.CharField(*max\_length*=50)

    def \_\_str\_\_(*self*):

        return *self*.cat\_name

    def get\_absolute\_url(*self*):

        return reverse("category\_detail", *kwargs*={"pk": *self*.pk})

class Product(*models*.Model):

    pID = models.AutoField(*primary\_key*=True)

    book\_name = models.CharField(*max\_length*=100)

    catID = models.ForeignKey(Category, *on\_delete*=models.CASCADE, *null*=True)

    book\_img = models.ImageField(

*upload\_to*='uploads/', *height\_field*=None, *width\_field*=None, *max\_length*=None, *null*=True, *blank*=True)

    book\_price = models.IntegerField(*default*=0, *null*=False)

    book\_stock = models.IntegerField(*default*=0, *null*=False)

    book\_description = models.TextField(*null*=True)

    def \_\_str\_\_(*self*):

        return str(*self*.pID)

    def get\_absolute\_url(*self*):

        # return f'/{self.slug}'

        return reverse("store:product\_detail", *kwargs*={"pID": *self*.pk})

    @property

    def get\_author(*self*):

        author\_list = list(*self*.author\_set.all())

        return author\_list

    @property

    def get\_imgURL(*self*):

        try:

            url = *self*.book\_img.url

        except:

            url = ''

        return url

class Author(*models*.Model):

    auID = models.AutoField(*primary\_key*=True)

    pID = models.ForeignKey(Product, *on\_delete*=models.CASCADE, *null*=True)

    au\_name = models.CharField(*max\_length*=50)

    def \_\_str\_\_(*self*):

        return *self*.au\_name

    def get\_absolute\_url(*self*):

        return reverse("store:author\_detail", *kwargs*={"auID": *self*.pk})

    @property

    def get\_list\_product(*self*):

*list* = Author.objects.filter(*auID*=*self*.auID)

        return *list*

class Customer(*models*.Model):

    cusID = models.AutoField(*primary\_key*=True)

    user = models.OneToOneField(

        User, *null*=True, *blank*=True, *on\_delete*=models.CASCADE)

    cus\_name = models.CharField(*max\_length*=50, *null*=True)

    cus\_addr = models.CharField(*max\_length*=50, *null*=True)

    cus\_phone = models.CharField(*max\_length*=12, *null*=True)

    def \_\_str\_\_(*self*):

        return *self*.cus\_name

    def get\_absolute\_url(*self*):

        return reverse("customer\_detail", *kwargs*={"pk": *self*.pk})

class Invoice(*models*.Model):

    iID = models.AutoField(*primary\_key*=True)

    cusID = models.ForeignKey(Customer, *on\_delete*=models.CASCADE, *null*=False)

    # date create

    date = models.DateField(*auto\_now*=False, *auto\_now\_add*=True, *null*=True)

    # checkout info

    date\_checkout = models.DateField(

*null*=True, *auto\_now*=False, *auto\_now\_add*=False)

    place\_status = models.BooleanField(

*null*=True, *default*=False)  # đã đặt hàng chưa? 0: chưa đặt

    # giao chưa? 0: giao chưa đến

    status = models.BooleanField(*null*=True, *default*=False)

    ship\_addr = models.CharField(*null*=True, *max\_length*=150)

    def \_\_str\_\_(*self*):

        return str(*self*.cusID.cus\_name)

    def get\_absolute\_url(*self*):

        return reverse("store:checkout\_detail", *kwargs*={"iID": *self*.pk})

    @property

    def get\_cus\_name(*self*):

        return *self*.cusID.cus\_name

    @property

    def get\_date\_checkout(*self*):

        return *self*.date\_checkout

    @property

    def get\_total\_item(*self*):

        orders = *self*.orderitem\_set.all()

        total = sum([i.quantity for i in orders])

        return total

    @property

    def get\_total\_price(*self*):

        orders = *self*.orderitem\_set.all()

        total = sum([i.get\_total for i in orders])

        return total

class OrderItem(*models*.Model):

    oID = models.AutoField(*primary\_key*=True)

    pID = models.ForeignKey(Product, *on\_delete*=models.CASCADE, *null*=False)

    iID = models.ForeignKey(Invoice, *on\_delete*=models.CASCADE, *null*=True)

    quantity = models.PositiveSmallIntegerField(*default*=0, *null*= True)

    # danh gia sp cua khach da mua hang

    comment = models.TextField(*null*=True, *max\_length*=254)

    @property

    def get\_total(*self*):

        total = *self*.pID.book\_price \* *self*.quantity

        return total

    def \_\_str\_\_(*self*):

        return str(*self*.oID)

    def get\_absolute\_cmt\_url(*self*):

        return reverse("store:order\_comment", *kwargs*={"oID": *self*.pk})

Sau khi cài đặt dùng lệnh “*py manage.py makemigrations*” và “*py manage migrate*” để ánh xạ vào CSDL.

Sau khi được ánh xạ ta có thể kiểm tra các bảng được tạo trong CSDL:

Graphical user interface, text, application, chat or text message

Description automatically generated

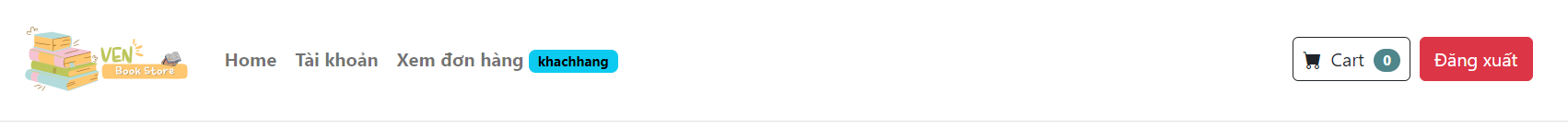
### 5.2. Cài đặt giao diện

Graphical user interface, text, application

Description automatically generated

Giao diện được chứa trong thư mục templates

#### 5.2.1. Navigation bar (Thanh điều hướng) và những thành phần hỗ trợ



Hình : Thanh điều hướng

Vị trí: \store\templates\includes\header.html

{% load static %}

<header>

  <nav *class*="navbar navbar-expand-lg navbar-light fixed-top bg-white border-bottom border-1">

    <div *class*="container px-4 px-lg-0">

      <a *class*="navbar-brand" *href*="{% url 'store:home' %}">

        <img *class*="myImageClass" *src*="{% static 'img/logo400x200.png' %}" *alt*="logo img" *height*="80">

      </a>

      <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarSupportedContent"

*aria-controls*="navbarSupportedContent" *aria-expanded*="false" *aria-label*="Toggle navigation">

        <span *class*="navbar-toggler-icon"></span>

      </button>

      <div *class*="collapse navbar-collapse" *id*="navbarSupportedContent">

        <ul *class*="navbar-nav me-auto mt-2 mt-lg-0">

          <li *class*="nav-item">

            <a *class*="nav-link fw-bold" *href*="{% url 'store:home' %}">Home</a>

          </li>

          {% if user.username %}

          <li *class*="nav-item">

            <a *class*="nav-link fw-bold" *href*="{% url 'store:edit\_profile' %}">Tài khoản</a>

          </li>

          <li>

            <a *class*="nav-link fw-bold" *href*="{% url 'store:checkout\_info' %}">Xem đơn hàng

              <span *class*="badge text-bg-info">{{user.username}}</span>

            </a>

          </li>

          {% endif %}

        </ul>

        <form *class*="d-flex" *action*="" *method*="POST">

          {% csrf\_token %}

          <a *href*="{% url 'store:cart' %}" *class*="btn btn-outline-dark px-2" *style*="border-radius: 5px;">

            <i *class*="bi-cart-fill me-1"></i>

            Cart

            <span *class*="badge bg-dark text-white ms-1 rounded-pill">{{invoice.get\_total\_item}}</span>

          </a>

          <div *class*="ps-2">

            {% if user.username %}

              <a *class*="btn btn-danger" *href*="{% url 'store:logout' %}" *style*="border-radius: 5px;">Đăng xuất</a>

            {% else %}

              <a *class*="btn btn-success" *href*="{% url 'store:login' %}" *style*="border-radius: 5px;">Đăng nhập</a>

            {% endif %}

          </div>

        </form>

      </div>

    </div>

  </nav>

</header>

##### Includes – Folder chứa những thành phần được sử dụng lặp lại

Những thành phần được chia nhỏ ra để đễ quản lý trong base.html bao gồm:

###### 1. Thành phần filter.html

Vị trí: store\templates\includes\filter.html

<!--included in index-->

<div *class*="col-2 border">

    <p>Thể loại: <strong *class*="text-info">{{choose}}</strong> </p>

    <form *action*="." *method*="POST">

      {% csrf\_token %}

      {% for cate in cates %}

        <div *class*="form-check">

          <input

*name*="catID"

*value*="{{cate.catID}}"

*class*="form-check-input"

*type*="radio"

*id*="{{cate.catID}}">

          <label *class*="form-check-label" *for*="{{cate.catID}}">

            {{cate.cat\_name}}

          </label>

        </div>

      {% endfor %}

      <div *class*="mt-2">

        <input *type*="submit" *value*="Lọc" *class*="btn btn-info rounded">

        <a *href*="{% url 'store:home' %}" *class*="btn btn-danger rounded">Xóa</a>

      </div>

    </form>

  </div>

###### 2. Thành phần header.html

Vị trí: \store\templates\includes\header.html

{% load static %}

<header>

  <nav *class*="navbar navbar-expand-lg navbar-light fixed-top bg-white border-bottom border-1">

    <div *class*="container px-4 px-lg-0">

      <a *class*="navbar-brand" *href*="{% url 'store:home' %}">

        <img *class*="myImageClass" *src*="{% static 'img/logo400x200.png' %}" *alt*="logo img" *height*="80">

      </a>

      <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarSupportedContent"

*aria-controls*="navbarSupportedContent" *aria-expanded*="false" *aria-label*="Toggle navigation">

        <span *class*="navbar-toggler-icon"></span>

      </button>

      <div *class*="collapse navbar-collapse" *id*="navbarSupportedContent">

        <ul *class*="navbar-nav me-auto mt-2 mt-lg-0">

          <li *class*="nav-item">

            <a *class*="nav-link fw-bold" *href*="{% url 'store:home' %}">Home</a>

          </li>

          {% if user.username %}

          <li *class*="nav-item">

            <a *class*="nav-link fw-bold" *href*="{% url 'store:edit\_profile' %}">Tài khoản</a>

          </li>

          <li>

            <a *class*="nav-link fw-bold" *href*="{% url 'store:checkout\_info' %}">Xem đơn hàng

              <span *class*="badge text-bg-info">{{user.username}}</span>

            </a>

          </li>

          {% endif %}

        </ul>

        <form *class*="d-flex" *action*="" *method*="POST">

          {% csrf\_token %}

          <a *href*="{% url 'store:cart' %}" *class*="btn btn-outline-dark px-2" *style*="border-radius: 5px;">

            <i *class*="bi-cart-fill me-1"></i>

            Cart

            <span *class*="badge bg-dark text-white ms-1 rounded-pill">{{invoice.get\_total\_item}}</span>

          </a>

          <div *class*="ps-2">

            {% if user.username %}

              <a *class*="btn btn-danger" *href*="{% url 'store:logout' %}" *style*="border-radius: 5px;">Đăng xuất</a>

            {% else %}

              <a *class*="btn btn-success" *href*="{% url 'store:login' %}" *style*="border-radius: 5px;">Đăng nhập</a>

            {% endif %}

          </div>

        </form>

      </div>

    </div>

  </nav>

</header>

###### 3. Thành phần message.html

{% if messages %}

{% for message in messages %}

<div >

    {% if message.level == DEFAULT\_MESSAGE\_LEVELS.SUCCESS %}

        <div *class*="alert alert-success alert-dismissible" *role*="alert">

            <div>

                {{message}}

            </div>

            <button *type*="button" *class*="btn-close" *data-bs-dismiss*="alert" *aria-label*="Close"></button>

        </div>

    {% endif %}

</div>

{% endfor %}

{% endif %}

###### 4. Thành phần search\_bar.html

<div *class*="container mb-3" *style*="width: 400px">

  <form *action*="{% url 'store:search' %}" *method*="post">

    {% csrf\_token %}

    <div *class*="input-group rounded">

      <input

*name*="search\_query"

*type*="search"

*class*="form-control rounded"

*placeholder*="Tìm kiếm"

      />

      <button *type*="submit" *class*="btn btn-outline-primary ms-2">

        <i *class*="fas fa-search"></i>

      </button>

    </div>

  </form>

</div>

#### 5.2.2. Trang chủ

Graphical user interface, application

Description automatically generated

Hình :Giao diện Trang chủ

Vị trí: \store\templates\home\index.html

{% extends 'store/base.html' %}

{% load static %}

{% block content %}

<div *class*="container">

    <div *class*="row row-cols-1 row-cols-md-4 g-4">

        {% for book in books %}

        <div *class*="col">

          <div *class*="card h-100">

            <a *href*="{{ book.get\_absolute\_url }}">

                <img *style*="" *src*="{{book.get\_imgURL}}" *class*="card-img-top" *alt*="(Không có ảnh sản phẩm)">

            </a>

            <div *class*="card-body text-center">

              <h5 *class*="card-title">{{book.book\_name}}</h5>

              <div *class*="d-flex justify-content-end">

                <h5 *class*="card-title">{{book.book\_price}}đ</h5>

              </div>

              <hr>

                <button

*class*="card-link btn btn-outline-success update-cart"

*data-product*={{book.pID}}

*data-action*="add">Thêm vào giỏ

                </button>

              <a *class*="card-link btn btn-outline-success" *href*="{{ book.get\_absolute\_url }}">View</a>

            </div>

            <div *class*="card-footer">

              <small *class*="text-muted">Last updated 3 mins ago</small>

            </div>

          </div>

        </div>

        {% endfor %}

    </div>

</div>

{% endblock content %}

#### 5.2.3. Giỏ hàng

Graphical user interface, text, application, email

Description automatically generated

Hình : Giao diện Giỏ hàng

Vị trí: \store\templates\home\cart\cart.html

{% extends 'layouts/base.html' %} {% load static %} {% load humanize %}

{% block content %}

<div *class*="row">

  <div *class*="col-lg-12">

    <div *class*="box-element">

      <a *class*="btn btn-outline-dark" *href*="{% url 'store:home' %}">&#x2190;Quay lại</a>

      <br />

      <br />

      <table *class*="table">

        <tr>

          <th>

            <h5>Số lượng: <strong>{{invoice.get\_total\_item}}</strong></h5>

          </th>

          <th>

            <h5>Tổng cộng: <strong>{{invoice.get\_total\_price|intcomma}}đ</strong></h5>

          </th>

          <th>

            <a *style*="float: right; margin: 5px" *class*="btn btn-success" *href*="{% url 'store:checkout' %}" >Đặt hàng</a>

          </th>

        </tr>

      </table>

    </div>

    <br />

    <div *class*="box-element">

      <div *class*="cart-row">

        <div *style*="flex: 2"></div>

        <div *style*="flex: 2"><strong>Sản phẩm</strong></div>

        <div *style*="flex: 1"><strong>Giá</strong></div>

        <div *style*="flex: 1"><strong>Số lượng</strong></div>

        <div *style*="flex: 1"><strong>Tổng cộng</strong></div>

        <div *style*="flex: 1"><strong>Xóa</strong></div>

      </div>

      {% for i in orders %}

      <div *class*="cart-row">

        <div *style*="flex: 2">

          <img *class*="row-image" *src*="{{i.pID.book\_img.url}}" />

        </div>

        <div *style*="flex: 2"><p>{{i.pID.book\_name}}</p></div>

        <div *style*="flex: 1"><p>{{i.pID.book\_price}}</p></div>

        <div *style*="flex: 1">

          <p *class*="quantity">{{i.quantity}}</p>

          <div *class*="quantity">

            <img *data-product*="{{i.pID.pID}}" *data-action*="add" *class*="chg-quantity update-cart" *src*="{% static 'img/arrow-up.png' %}"/>

            <img *data-product*="{{i.pID.pID}}" *data-action*="remove" *class*="chg-quantity update-cart" *src*="{% static 'img/arrow-down.png' %}"/>

          </div>

        </div>

        <div *style*="flex: 1"><p>{{i.get\_total}}</p></div>

        <div *style*="flex: 1">

          <a *data-product*="{{i.pID.pID}}" *data-action*="delete" *class*="chg-quantity update-cart btn-close" *type*="button" ></a>

        </div>

      </div>

      {% endfor %}

    </div>

  </div>

</div>

{% endblock content %}

#### 5.2.4. Đặt hàng

Graphical user interface, text, application

Description automatically generated

Hình : Giao diện đặt hàng

Vị trí: \store\templates\home\checkout\checkout.html

{% extends 'layouts/base.html' %} {% load static %} {% load humanize %}

{% block content %}

<div *class*="row">

  <a *class*="btn btn-outline-dark" *href*="{% url 'store:cart' %}"

    >&#x2190; Quay về giỏ hàng</a

  >

  {% if user.username %} {% if orders %}

  <div *class*="col-lg-6">

    <div *class*="container box-element">

      <form *action*="{% url 'store:checkout\_submit' %}" *method*="POST">

        {% csrf\_token %}

        <h2 *class*="text-center">Thông tin giao nhận</h2>

        <div *class*="mb-3 mt-3">

          <label *for*="ship\_addr" *class*="form-label">Địa chỉ nhận hàng:</label>

          <input

*id*="ship\_addr" *name*="ship\_addr" *value*="{{invoice.cusID.cus\_addr}}"

*type*="text" *class*="form-control" *placeholder*="Nhập địa chỉ nhận hàng"

          />

        </div>

        <input *type*="hidden" *name*="iID" *value*="{{invoice.iID}}" />

        <button *type*="submit" *class*="btn btn-primary">Đặt hàng</button>

      </form>

    </div>

  </div>

  {% else %}

  <div *class*="col-lg-6">

    <div *class*="alert alert-danger" *role*="alert">

      Giỏ hàng của bạn đang trống

      <a *href*="{% url 'store:home' %}" *class*="alert-link">quay lại</a> để mua hàng!

    </div>

  </div>

  {% endif %}

  <div *class*="col-lg-6">

    <div *class*="box-element">

      <p>Số lượng sản phẩm: <strong>{{invoice.get\_total\_item}}</strong></p>

      <p>Tổng cộng: <strong>{{invoice.get\_total\_price|intcomma}}đ</strong></p>

      <div *class*="cart-row">

        <div *style*="flex: 2"><img *class*="row-image" *src*="" /></div>

        <div *style*="flex: 2">

          <p>Tên sách</p>

        </div>

        <div *style*="flex: 1">

          <p>Giá</p>

        </div>

        <div *style*="flex: 1">

          <p>Số lượng</p>

        </div>

      </div>

      {% for i in orders %}

      <div *class*="cart-row">

        <div *style*="flex: 2">

          <img *class*="row-image" *src*="{{ i.pID.book\_img.url}}" />

        </div>

        <div *style*="flex: 2">

          <p>{{i.pID.book\_name}}</p>

        </div>

        <div *style*="flex: 1">

          <p>{{i.pID.book\_price|intcomma}}đ</p>

        </div>

        <div *style*="flex: 1">

          <p>{{i.quantity}}</p>

        </div>

      </div>

      {% endfor %}

    </div>

  </div>

  <!-- ========== Start Thông báo cho khách nếu chưa đăng nhập ========== -->

  {% else %}

  <div *class*="alert alert-danger" *role*="alert">

    Bạn cần

    <a *href*="{% url 'store:login' %}" *class*="alert-link">đăng nhập</a> hoặc

    <a *href*="{% url 'store:register' %}" *class*="alert-link">đăng ký</a> để mua hàng!

  </div>

  {% endif %}

  <!-- ========== End Thông báo cho khách nếu chưa đăng nhập ========== -->

</div>

{% endblock content %}

#### 5.2.5. Giao diện khách hàng chỉnh sửa thông tin cá nhân

Graphical user interface, application

Description automatically generated

Hình : Giao diện chỉnh sửa thông tin

#### 5.2.6. Giao diện admin của django

Khi đăng nhập vào [*http://127.0.0.1:8000/admin*](http://127.0.0.1:8000/admin) *với tài khoản admin:*

A screenshot of a computer

Description automatically generated with medium confidence

Ở giao diện admin người dùng (quản lý) có quyền chỉnh sửa trên toàn hệ thống

### 5.3. Các hàm xử lý chính

#### 5.3.1 Xử lí thêm, sửa, xóa sản phẩm

Vị trí: “.\bookstore\static\js\cart.js”

var updateBtn = document.getElementsByClassName("update-cart");

for (var i = 0; i < updateBtn.length; i++) {

  updateBtn[i].addEventListener("click", function () {

    var productID = *this*.dataset.product;

    //data-product={{i.pID}} data-action="add"

    var action = *this*.dataset.action;

    console.log("productID:", productID, "action:", action);

    console.log("USER:", user);

    if (user == "AnonymousUser") {

      console.log("User is not authenticated");

      alert("Bạn phải đăng nhập để mua hàng!!");

    } else {

      //console.log('productID:', productID, 'action:', action)

      //console.log('USER:', user)

      updateUserOrder(productID, action);

    }

  });

}

function updateUserOrder(*productID*, *action*) {

  console.log("(updateUserOrder) User is authenticated, sending data...");

  var url = "http://127.0.0.1:8000/update\_item/";

  //console.log(url)

  fetch(url, {

    method: "POST",

    headers: {

      "Content-Type": "application/json",

      "X-CSRFToken": csrftoken,

    },

    body: JSON.stringify({ productID: *productID*, action: *action* }),

  })

    .then((*response*) => {

      return *response*.json();

    })

    .then((*data*) => {

      console.log("data:", *data*);

      location.reload();

    });

}

#### 5.3.2 Xử lí đặt hàng

Vị trí: “.\bookstore\static\js\checkout.js”

def checkout\_submit(*request*):

    iID = *request*.POST['iID']

    ship\_addr = *request*.POST['ship\_addr']

    try:

        i = Invoice.objects.get(*iID*=iID)

        i.ship\_addr = ship\_addr

        i.date\_checkout = datetime.now()

        i.place\_status = True

        i.save()

        return redirect('store:checkout\_info')

    except:

        return HttpResponse("Đặt hàng không thành công")

# KẾT LUẬN

## 1. Kết quả đạt được sau khi thực hiện đề tài:

Sau thời gian em tìm hiểu và thực hiện đề tài “xây dựng website bán sách” kết quả đạt được như sau:

* Trong quá trình thực hiện, khi tìm tài liệu bằng tiếng việt khá hạn chế nên việc tiếp xúc tài liệu tiếng anh không thể tránh khỏi đã giúp em cải thiện một ít trình độ nghe, đọc tiếng anh chuyên ngành.
* Hiểu biết thêm về các ngôn ngữ lập trình và framework:
* Đầu tiên là các ngôn ngữ dùng để viết website: HTML, CSS, JavaScript. Biết được cách sử dụng các thẻ HTML để đưa dữ liệu hiện thị cho người dùng, và sử dụng CSS để làm đẹp cho website, ngoài ra sử dụng JavaScript để tạo ra các tương tác với người dùng (như click chuột, di chuột ngang qua,…)
* Thứ hai là biết sử các class của Bootstrap: nhằm tăng hiệu quả của việc hiện thị thông tin đến người dùng, tăng tốc độ lập trình.
* Thứ ba là về cơ sở dữ liệu và hệ quản trị CSDL MySQL: Qua đợt làm đề tài đồ án này em đã hiểu rõ hơn về CSDL, ví dụ như cách thiết kế một CSDL dựa trên các yêu cầu cho trước; kiểu dữ liệu trong CSDL tương ứng với các kiểu dữ liệu trong ngôn ngữ lập trình như thế nào; Các đối tượng trong ngôn ngữ lập trình tương ứng với các bảng trong CSDL, Thêm sửa xóa dữ liệu trong database cần những điều kiện như: không thể xóa một bản ghi khi mà có các khóa liên kết
* Cuối cùng là biết thêm về python và framework Django, cách sử dụng và ứng dụng nó vào đề tài, cụ thể trong thời gian làm việc với django, mặc dù chỉ mới khai thác được một phần nhỏ của nó nhưng đã có thể giúp em hoàn chỉnh website của mình: sử dụng mô hình MVC (Model-View-Controller) vì bản thân django chính là sử dụng mô hình này để thiết kế web, ngoài ra django còn cung cấp một hệ thống quản trị người dùng vô cùng mạnh mẽ, bảo mật, biết cách sử dụng các models giúp em tương tác với CSDL một cách nhanh chóng, hiểu thêm về thừa kết trong kỹ thuật hướng đối tượng.

## 2. Ưu nhược điểm của website

### Ưu điểm:

* + - Giao diện đơn giản, dễ sử dụng
    - Đáp ứng được các yêu cầu cho người dùng như:
  + Đăng ký tài khoản (Có thể đăng nhập bằng tài khoản Google)
  + Đăng nhập
  + Xem và lọc sách
  + Thêm sách vào giỏ hàng
  + Kiểm tra giỏ hàng
  + Đặt hàng (thanh toán khi nhận hàng)

### Nhược điểm:

* + Hình thức thanh toán còn hạn chế (chỉ có thanh toán khi nhận hàng)
  + Giao diện không có nhiều hiệu ứng
  + Chức năng còn hạn chế

## 3. Các chức năng có thể phát triển

* Cải thiện giao diện đồ họa bằng các template
* Thông báo email khi đăng ký người dùng và đặt hàng
* Cho phép người dùng đặt hàng mà không cần đăng ký tài khoản
* Xác nhận email trước khi thực hiện thanh toán

# TÀI LIỆU THAM KHẢO

Cơ sở lý thuyết:

<https://baodautu.vn/mua-sam-online-tiep-tuc-la-xu-huong-trong-nam-2022-d159926.html> [1]

<https://ezweb.vn/blog/8-yeu-to-can-co-cua-mot-website-ban-hang-chuyen-nghiep.html> [2]

<https://www.duhoctrungquoc.vn/wiki/vi/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u_quan_h%E1%BB%87> [3]

<https://www.hostinger.vn/huong-dan/mysql-la-gi/> [4]

<https://aws.amazon.com/vi/what-is/python/> [5]

<https://viblo.asia/p/python-co-ban-voi-django-framework-Ljy5VxGkZra> [6]

[https://www.hostify.vn/blog/html-css-javascript-la-gi [7](https://www.hostify.vn/blog/html-css-javascript-la-gi%20%5b7)]

Thực hành:

<https://docs.djangoproject.com/en/4.1/>

<https://www.youtube.com/watch?v=_ELCMngbM0E&t=1s>

<https://pylessons.com/django-google-oauth>

<https://www.google.com/recaptcha/admin/site/596490496>

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

<https://www.w3schools.com/bootstrap5/index.php>

<https://howkteam.vn/course/lap-trinh-web-voi-python-bang-django/gioi-thieu-ve-python-django-1515>

<https://fontawesome.com/kits>