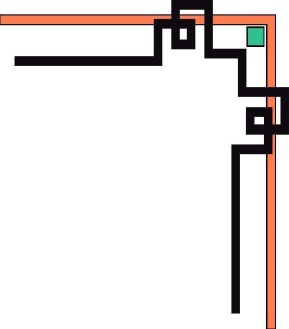
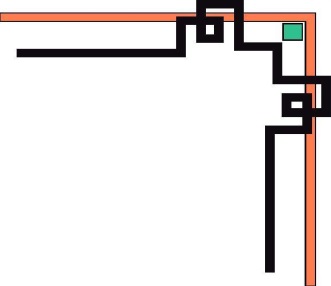
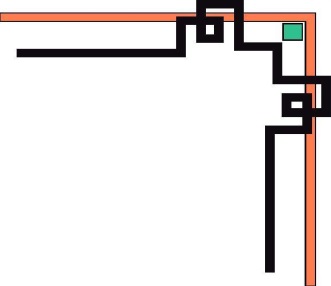
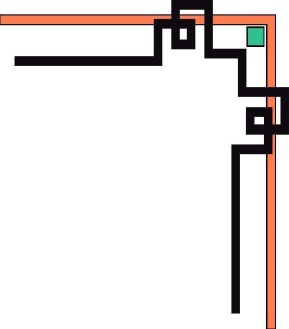
BỘ GIÁO DỤC VÀ ĐÀO TẠO



**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**

**BẰNG NGÔN NGỮ PYTHON 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](#_Toc121830852)

[DANH MỤC HÌNH ẢNH 3](#_Toc121830853)

[DANH MỤC BẢNG BIỂU 3](#_Toc121830854)

[NỘI DUNG 4](#_Toc121830855)

[PHẦN 1: ĐẶC TẢ 4](#_Toc121830856)

[PHẦN 2: CÔNG VIỆC 5](#_Toc121830857)

[PHẦN 3: CƠ SỞ LÝ THUYẾT 6](#_Toc121830858)

[3.1. Cơ sở dữ liệu quan hệ là gì? 6](#_Toc121830859)

[3.2. MySQL là gì? 6](#_Toc121830860)

[3.3. Python và Django 7](#_Toc121830861)

[3.4. HTML, CSS, JavaScript là gì? 9](#_Toc121830862)

[PHẦN 4: THIẾT KẾ 11](#_Toc121830863)

[4.1. Thiết kế CSDL 11](#_Toc121830864)

[4.2. Thiết kế giao diện 13](#_Toc121830865)

[4.3. Một số phương thức xử lí trong models 14](#_Toc121830866)

[PHẦN 5: CÀI ĐẶT VÀ KIỂM THỬ 15](#_Toc121830867)

[5.1. Cài đặt các models 15](#_Toc121830868)

[5.2. Cài đặt giao diện 18](#_Toc121830869)

[5.3. Xử lí thêm, sửa, xóa sản phẩm 24](#_Toc121830870)

[5.4. Xử lí đặt hàng 25](#_Toc121830871)

[KẾT LUẬN 27](#_Toc121830872)

[1. Kết quả đạt được: 27](#_Toc121830873)

[2. Các hạn chế: 27](#_Toc121830874)

[3. Các chức năng có thể phát triển 27](#_Toc121830875)

[TÀI LIỆU THAM KHẢO 28](#_Toc121830876)

# DANH MỤC HÌNH ẢNH

[Hình 1: Chức năng chính của website 7](#_Toc121830639)

[Hình 2: Mô hình ERD 13](#_Toc121830640)

[Hình 3: Thanh điều hướng 21](#_Toc121830641)

[Hình 4:Giao diện Trang chủ 23](#_Toc121830642)

[Hình 5: Giao diện Giỏ hàng 24](#_Toc121830643)

[Hình 6: Giao diện đặt hàng 26](#_Toc121830644)

# DANH MỤC BẢNG BIỂU

[Bảng 1: Bảng mô tả các trường dữ liệu của models 14](#_Toc121830736)

[Bảng 2: Bảng thiết kế giao diện 15](#_Toc121830737)

[Bảng 3: Một số phương thức xử lí 16](#_Toc121830738)

[Bảng 4: Các lệnh cơ bản trong django 17](#_Toc121830739)

# NỘI DUNG

## PHẦN 1: ĐẶC TẢ

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)

Để đáp ứng được những nhu cầu trên, thiết kế một website thương mại điện tử nhằm phục vụ việc mua sắm online là rất hợp lý trong thời điểm này. Và để thiết kế một website nhanh chóng và hiệu quả thì sử dụng Framework Django là một trong những lựa chọn tốt nhất hiện tại.

Các **tính năng chính** của trang web:

* Đăng ký tài khoản, Đăng nhập, Đăng xuất
* Xem tất cả sản phẩm (sách), khi ấn vào sách có thể xem chi tiết (số lượng, mô tả ủa sản phẩm, đánh giá, tác giả)
* Thêm vào giỏ hàng, chỉnh sửa số lượng trong giỏ hàng (tăng, giảm, xóa)
* Thanh toán khi nhận hàng (cần nhập địa chỉ)

## PHẦN 2: CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **Công việc** | Chi tiết | Hoàn thành |
| Thiết kế CSDL |  | 100% |
| Thiết kế giao diện | * + Trang chủ   + Trang giỏ hàng   + Trang thanh toán   + Giao diện chức năng khác | 100% |
| Xử lý các chức năng chính | [Chi tiết](#Chucnangchinhcuawebsite) | 100% |



Hình 1: Chức năng chính của website

## PHẦN 3: CƠ SỞ LÝ THUYẾT

### 3.1. Cơ sở dữ liệu quan hệ là gì?

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?

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

* Các nhà phát triển có thể dễ dàng đọc và hiểu một chương 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 chương 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?

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.

**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.

### 3.4. HTML, CSS, JavaScript là gì?

**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.

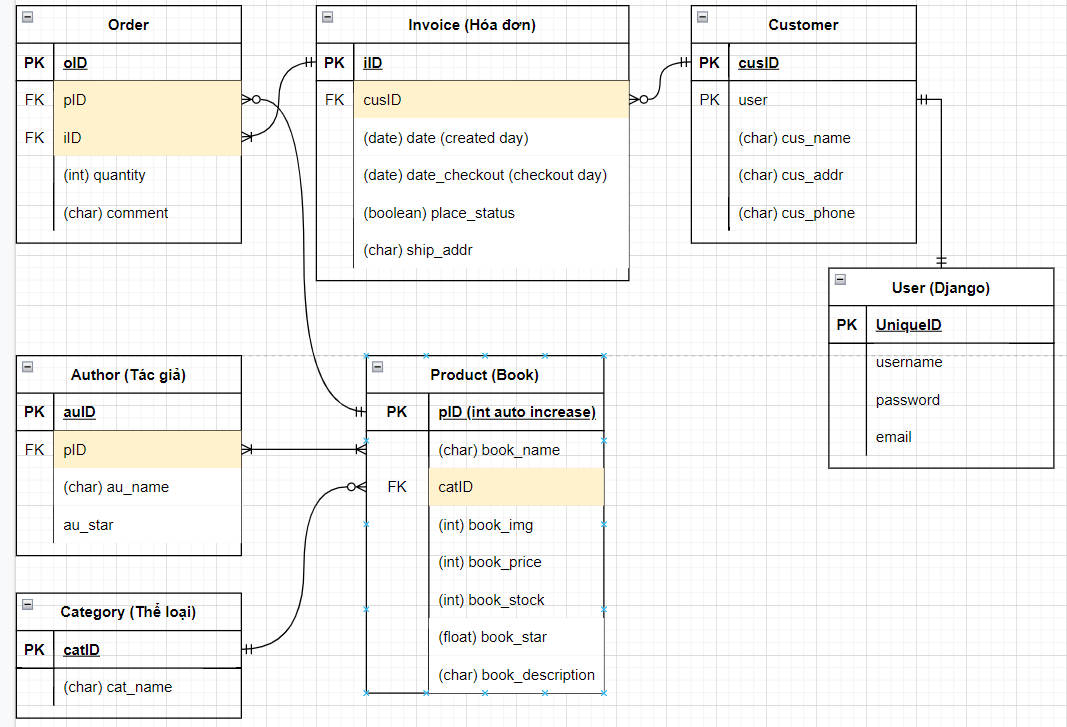
Đây là 3 ngôn ngữ chính trong lập trình web và có mối liên hệ lẫn nhau để xây dựng một website sống động, chuyên nghiệp:

* 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.

## PHẦN 4: THIẾT KẾ

### 4.1. Thiết kế CSDL

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



Hình 2: Mô hình ERD

#### Mô tả các trường của CSDL

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

Bảng 1: Bảng mô tả các trường dữ liệu của models

### 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 (chưa hoàn thành)* * Giỏ hàng để vào giao diện giỏ hàng |
| 2 | Store.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. |

Bảng 2: Bảng thiết kế giao diện

### 4.3. Một số phương thức xử lí trong models

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên | Vị trí | Mục đích |
| 1 | get\_total(*self*) | store/models.py  dòng 132 | Tính tổng giá của một order.  Vd: order 2 cuốn sách 50k  Kết quả trả về: 100k |
| 2 | get\_total\_item(*self*) | 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 | get\_total\_price(*self*) | 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 |

Bảng 3: Một số phương thức xử lí

## PHẦN 5: CÀI ĐẶT VÀ KIỂM THỬ

|  |  |  |
| --- | --- | --- |
| 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. |

Bảng 4: Các lệnh cơ bản trong django

### 5.1. Cài đặt các models

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*=50)

    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\_star = models.FloatField(*null*=True)

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

    def \_\_str\_\_(*self*):

        return str(*self*.pID)

    def get\_absolute\_url(*self*):

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

    @property

    def get\_author(*self*):

        aus = *self*.author\_set.all()

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

        for i in author\_list:

            print(i)

        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)

    au\_star = models.FloatField(*default*=0)

    def \_\_str\_\_(*self*):

        return *self*.au\_name

    def get\_absolute\_url(*self*):

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

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)

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

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

    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

    status = models.BooleanField(*null*=True, *default*=False) #giao chưa? 0: giao chưa đến

    ship\_addr = models.CharField(*null*=True,*max\_length*=254)

    def \_\_str\_\_(*self*):

        return str(*self*.iID)

    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*.order\_set.all()

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

        return total

    @property

    def get\_total\_price(*self*):

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

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

        return total

class Order(*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.IntegerField(*null*=False, *default*=0)

    #danh gia sp cua khach da mua hang

    comment = models.TextField(*null*=True)

    @property

    def get\_total(*self*):

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

        return total

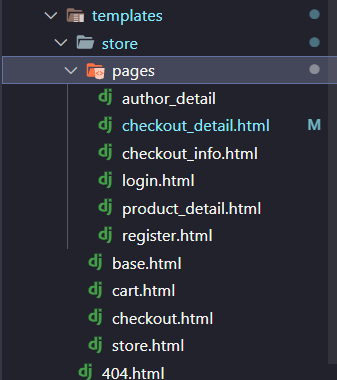
    def \_\_str\_\_(*self*):

        return *self*.oID

    def get\_absolute\_url(*self*):

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

### 5.2. Cài đặt giao diện



Giao diện được chứa trong thư mục templates

#### 5.2.1. Navigation bar (Thanh điều hướng)



Hình 3: Thanh điều hướng

Vị trí: trong file base.html

<header>

    <!-- ========== Start navigation bar ========== -->

    <nav *class*="navbar navbar-expand-lg navbar-light">

      <div *class*="container px-4 px-lg-5">

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

          <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 active" *href*="{% url 'store:store' %}" *aria-current*="page">Home <span

*class*="visually-hidden">(current)</span></a>

            </li>

            <li *class*="nav-item dropdown">

              <a *class*="nav-link dropdown-toggle" *href*="#" *id*="dropdownId" *data-bs-toggle*="dropdown"

*aria-haspopup*="true" *aria-expanded*="false">Tài khoản</a>

              <div *class*="dropdown-menu" *aria-labelledby*="dropdownId">

                {% if user.username %}

                  <a *class*="btn btn-danger" *href*="{% url 'store:logout' %}">Đăng xuất</a>

                <br>

                <a *class*="btn btn-info" *href*="{% url 'store:checkout\_info' %}">Đơn hàng</a>

                {% else %}

                <p>

                  <a *class*="btn btn-info" *href*="{% url 'store:register' %}">Đăng ký</a>

                  <br>

                  <a *class*="btn btn-success" *href*="{% url 'store:login' %}">Đăng nhập</a>

                </p>

                {% endif %}

              </div>

            </li>

            <li>

              {% if user.username %}

              <span *class*="badge text-bg-primary fs-6">Xin chào {{user.username}}</span>

              {% else %}

              <span *class*="badge text-bg-danger fs-6">Bạn chưa đăng nhập</span>

              {% endif %}

            </li>

          </ul>

          <form *class*="d-flex">

            <input *class*="px-2" *type*="text" *placeholder*="Tìm kiếm" *style*="border-radius: 5px;">

            <button *class*="btn btn-success mx-2" *type*="submit" *style*="border-radius: 5px;">Search</button>

            <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>

          </form>

        </div>

      </div>

    </nav>

    <!-- ========== End navigation bar ========== -->

  </header>

#### 5.2.2. Trang chủ

Graphical user interface, application

Description automatically generated

Hình 4:Giao diện Trang chủ

Vị trí: trong file store.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, application

Description automatically generated

Hình 5: Giao diện Giỏ hàng

Vị trí: trong file cart.html

{% extends 'store/base.html' %} {% load static %} {% block content %}

<div *class*="row">

  <div *class*="col-lg-12">

    <div *class*="box-element">

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

      <br />

      <br />

      <table *class*="table">

        <tr>

          <th>

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

          </th>

          <th>

            <h5>Total: <strong>{{invoice.get\_total\_price}}đ</strong></h5>

          </th>

          <th>

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

Description automatically generated

Hình 6: Giao diện đặt hàng

Vị trí: trong file checkout.html

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

{% load static %}

{% block content %}

<div *class*="row">

    {% if user.username %}

        {% if orders %}

        <div *class*="col-lg-6">

            <div *class*="container mt-3 box-element" >

                <form *action*="" *id*="form" *data-invoice*="{{invoice.iID}}" *data-action*="place" >

                    <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>

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

                </div>

            </div>

        {% endif %}

    <div *class*="col-lg-6">

        <div *class*="box-element">

            <a  *class*="btn btn-outline-dark" *href*="{% url 'store:cart' %}">&#x2190; Back to Cart</a>

            <hr>

            <h3>Order Summary</h3>

            <hr>

            <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}}đ</p></div>

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

            </div>

            {% endfor %}

            <h5>Items: {{invoice.get\_total\_item}}</h5>

            <h5>Total: {{invoice.get\_total\_price}}đ</h5>

        </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>

<script *src*="{% static 'js/checkout.js' %} "></script>

{% endblock content %}

### 5.3. 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();

    });

}

### 5.4. Xử lí đặt hàng

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

var form = document.getElementById("form");

form.addEventListener("submit", function () {

  var addr = document.getElementById("ship\_addr").value;

  var iID = *this*.dataset.invoice;

  //data-place\_addr={{i.pID}} data-action="place"

  var action = *this*.dataset.action;

  console.log("place\_addr:", addr, "action:", action, "iID:", iID);

  console.log("USER:", user);

  if (user == "AnonymousUser") {

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

  } else {

    placeOrder(addr, action, iID);

    alert\_success();

    back\_home();

  }

});

function placeOrder(*addr*, *action*, *iID*) {

  console.log("Place order");

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

  var url = "http://127.0.0.1:8000/place/";

  fetch(url, {

    method: "POST",

    headers: {

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

      "X-CSRFToken": csrftoken,

    },

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

  })

    .then((*response*) => {

      return *response*.json();

    })

    .then((*data*) => {

      console.log("data:", *data*);

      //console.debug;

      location.reload()

    });

}

function alert\_success() {

  alert("Bạn đã đặt hàng thành công");

}

function back\_home() {

  location.replace("http://127.0.0.1:8000/");

}

# KẾT LUẬN

## 1. Kết quả đạt được:

* Xây dựng được cơ bản website bán hàng
  + - Giao diện đơn giản, dễ sử dụng
    - Đáp ứng các nhu cầu cơ bản người dùng
* Biết thêm về HTML, CSS, JavaScript, Bootstrap, Django, Thiết kế CSDL, Hệ quản trị CSDL MySQL.

## 2. Các hạn chế:

* + Thanh toán khi giao hàng
  + Giao diện đồ họa đơn giản
  + Chức năng còn hạn chế

## 3. Các chức năng có thể phát triển

* Thông báo email khi đăng ký người dùng và đặt hàng
* Chức năng tìm kiếm sản phẩm

# TÀI LIỆU THAM KHẢO

[Một số ý kiến về View của Django](https://viblo.asia/p/mot-so-y-kien-ve-view-cua-django-Do754EzWKM6)

[Python cơ bản với Django Framework](https://viblo.asia/p/python-co-ban-voi-django-framework-Ljy5VxGkZra)

[HTML CSS Javascript là gì? Giải đáp 3 câu hỏi thường gặp](https://www.hostify.vn/blog/html-css-javascript-la-gi/)

[JavaScript là gì? Kiến thức cơ bản về JavaScript cho người mới bắt đầu 2022 - Tin tức tên miền hosting](https://tenten.vn/tin-tuc/javascript-la-gi/#JavaScript_la_gi)

[Lập trình web với Python bằng Django | How Kteam](https://howkteam.vn/course/lap-trinh-web-voi-python-bang-django-36)

[(1) Django Ecommerce Website | Setup & Templates | Part 1 - YouTube](https://www.youtube.com/watch?v=_ELCMngbM0E&t=1s)

[CodeWithSteps](https://codewithsteps.herokuapp.com/project/cd0492f3-ee93-471a-9dbc-b047233336c3/)

[Django overview | Django](https://www.djangoproject.com/start/overview/)

[Django Tutorial](https://www.w3schools.com/django/)

[Font Awesome](https://fontawesome.com/kits)