

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐÔNG Á**



**ĐỒ ÁN CHUYÊN NGÀNH
PYTHON DJANGO**

**Đề tài:
Shop bán hàng**

**GVHD: TS. Nguyễn Sĩ Thìn
SVTH: Lương Quang Hùng
Lớp: ST21A1A**

Đà Nẵng, tháng 04/2024

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn các thầy, cô trong khoa Công Nghệ Thông Tin cũng như toàn thể mọi người trong ngôi trường Đại học Đông Á đã tạo điều kiện thuận lợi cho em trong suốt quá trình học tập tại trường cũng như trong thời gian thực hiện đồ án tốt nghiệp. Đặc biệt, em muốn gửi lời cảm ơn tới Tiến Sĩ – Nguyễn Sĩ Thìn giảng viên trực tiếp hướng dẫn tận tình chỉ bảo giúp em khắc phục những khó khăn, thiếu sót để có thể hoàn thành các phần trong đồ án tốt nghiệp từ lý thuyết cho tới thực hành sử dụng công cụ. Với hiểu biết tìm tòi của bản thân và sự chỉ bảo hướng dẫn tận tình của giảng viên em đã cố gắng hoàn thành đồ án một cách tốt nhất có thể nhưng cũng không thể tránh được thiếu sót. Kính mong nhận được sự đóng góp ý kiến từ thầy cô để em có thể nâng cao cũng như bổ sung thêm kiến thức cho bản thân, hoàn thiện đồ án với một kết quả tốt và hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

MỞ ĐẦU

Trong cuộc sống của mỗi chúng ta khi nền kinh tế ngày càng phát triển, ngành công nghệ thông tin trở thành một trợ thủ, điều không thể thiếu với hầu hết các ngành nghề. Có thể ban đầu nhiều người sẽ cảm thấy chưa cần thiết vì chỉ kinh doanh ở quy mô nhỏ, hoặc chưa sẵn sàng sử dụng vì nghĩ rằng phần mềm là một khái niệm gì đó rất mới mẻ, phức tạp, khó sử dụng. Ngày nay trong cuộc sống 4.0 của chúng ta gần như không thể thiếu được những khái niệm liên quan đến công nghệ thông tin, từ các phương tiện truyền thông, xã hội cho đến lĩnh vực kinh doanh, quản lý.... Mọi thứ đều cần những phần mềm cũng như sản phẩm của công nghệ thông tin để hỗ trợ giúp nâng cao phát triển vững mạnh hơn. Ví dụ trong lĩnh vực kinh doanh buôn bán hàng hóa theo cách truyền thống vẫn còn tồn tại nhiều nhược điểm như thống kê chi tiết chưa chính xác, quy trình thanh toán chưa được chặt chẽ và nhanh chóng. Cùng với sự phát triển của nhiều ngôn ngữ lập trình web như PHP, Ruby, Scheme thì Python là một cái tên đáng chú ý. Hiện nay ngôn ngữ Python được xếp hạng thứ 1 trong Top 10 các ngôn ngữ lập trình phổ biến nhất đang được thế giới sử dụng. Python là một ngôn ngữ có hình thái rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới bắt đầu học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu nhất. Python là một ngôn ngữ lập trình đơn giản nhưng lại rất hiệu quả. Bên cạnh đó, Python là một ngôn ngữ có tính hướng đối tượng cao. Với ngôn ngữ lập trình python là một ngôn ngữ lập trình đa năng với nhiều ưu

điểm vượt trội, và đang đứng ở vị trí số một trong top các ngôn ngữ lập trình hiện nay.

MỤC LỤC

CHƯƠNG I: TỔNG QUAN VỀ PYTHON DJANGO.....	3
I. Giới thiệu về web framework django Django	3
1.1. Cài đặt django	3
1.2. Tạo project.....	4
1.3. Chạy server	5
1.4. Tạo web app.....	6
1.5. Model.....	7
1.6. Hệ thống ADMIN	10
1.7. View và Templates	14
1.8. Upload File.....	22
1.9. Form trong Django	32
1.10. Hệ thống User trong DjangoLỗi! Thẻ đánh dấu không được xác định.	
1.11. Custom User model trong DjangoLỗi! Thẻ đánh dấu không được xác định.	
1.12. Custom ADMIN site trong DjangoLỗi! Thẻ đánh dấu không được xác định.	
II. Xây dựng web site bán hàng....Lỗi! Thẻ đánh dấu không được xác định.	
2.1. Phân tích cơ sở dữ liệu.....Lỗi! Thẻ đánh dấu không được xác định.	
2.2. Xây dựng website bán hàng	53
CHƯƠNG III: Kết Luận	64

CHƯƠNG I: TỔNG QUAN VỀ PYTHON DJANGO

I. Giới thiệu về web framework django Django

Giới thiệu về web framework django. Django là một web framework khá nổi tiếng được viết hoàn toàn bằng ngôn ngữ Python. Nó là một framework với đầy đủ các thư viện, module hỗ trợ các web-developer. 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”, tính năng phát triển nhanh, không làm lại những gì đã làm. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy”, tính năng phát triển nhanh, không làm lại những gì đã làm. Django được thiết kế với triết lý làm sao để các lập trình viên đưa các ý tưởng trở thành một sản phẩm nhanh nhất có thể. Với sự kết hợp hoàn hảo đó chúng ta hoàn toàn có thể xây dựng một website bán hàng hay quản lý hàng hóa với độ chi tiết và chính xác cao.

Một số ưu điểm khi dùng Django là :

- Nhanh: Django được thiết kế với triết lý làm sao để các lập trình viên đưa các ý tưởng trở thành một sản phẩm nhanh nhất có thể.
- Có đầy đủ các thư viện/module cần thiết: Django có sẵn các thư viện về user authentication, content admin, site maps...
- Đảm bảo về tính bảo mật: Không còn các nỗi lo về các lỗi bảo mật thông thường. Django cũng cung cấp cả phương pháp để lưu mật khẩu an toàn .
- Khả năng mở rộng tốt: Django có thể đáp ứng lượng traffic lớn.

1.1. Cài đặt django

Django là framework giúp cho việc xây dựng các website và phát triển ứng dụng web một cách dễ dàng nhanh chóng hơn, và ít code hơn. Đầu tiên chúng ta truy cập vào trang chủ của django <https://www.djangoproject.com/download/> để tìm hiểu và download. Đầu tiên cần tải pip về để cài đặt Django. Chúng ta tải pips tại <https://pip.pypa.io>. Tiến hành download file get-pip.py về. Tiếp theo chúng ta bật cửa sổ CMD lên và chạy file get-pip.py..

```

PS V:\AppDjango> python -v
import _frozen_importlib # frozen
import _imp # builtin
import '_thread' # <class '_frozen_importlib.BuiltinImporter'>
import '_warnings' # <class '_frozen_importlib.BuiltinImporter'>
import '_weakref' # <class '_frozen_importlib.BuiltinImporter'>
import 'winreg' # <class '_frozen_importlib.BuiltinImporter'>
import '_io' # <class '_frozen_importlib.BuiltinImporter'>
import 'marshal' # <class '_frozen_importlib.BuiltinImporter'>
import 'nt' # <class '_frozen_importlib.BuiltinImporter'>
import '_frozen_importlib_external' # <class '_frozen_importlib.FrozenImporter'>
# installing zipimport hook
import 'time' # <class '_frozen_importlib.BuiltinImporter'>
import 'zipimport' # <class '_frozen_importlib.FrozenImporter'>
# installed zipimport hook
# C:\Python312\Lib\encodings\__pycache__\__init__.cpython-312.pyc matches C:\Python312\Lib\encodings\__init__
__py
# code object from 'C:\Python312\Lib\encodings\__pycache__\__init__.cpython-312.pyc'
import '_codecs' # <class '_frozen_importlib.BuiltinImporter'>
import 'codecs' # <class '_frozen_importlib.FrozenImporter'>

```

Python

Sau khi cài xong pips ta tiến hành download Django bằng cách gõ câu lệnh: “ pip install Django” để cài đặt django.

1.2. Tạo project

Bật cửa sổ CMD trên máy tính và gõ lệnh django-admin startproject Trong đó django-admin startproject là lệnh để tạo project , HungDjango là tên của project Chúng ta cũng có thể sử dụng môi trường lập trình PyCharm để làm việc .Sau khi đã cài đặt được django trên máy tính, trong cửa sổ terminal của PyCharm chúng ta sẽ tiến hành tạo project bằng lệnh “django-admin startproject HungDjango”.

```

PS V:\AppDjango\luong-quang-hung> python -m pip install Django
>>
Requirement already satisfied: Django in c:\python312\lib\site-packages (5.0.3)
Requirement already satisfied: asgiref<4,>=3.7.0 in c:\python312\lib\site-packages (from Django) (3.7.2)
Requirement already satisfied: sqlparse>=0.3.1 in c:\python312\lib\site-packages (from Django) (0.4.4)
Requirement already satisfied: tzdata in c:\python312\lib\site-packages (from Django) (2024.1)
PS V:\AppDjango\luong-quang-hung>

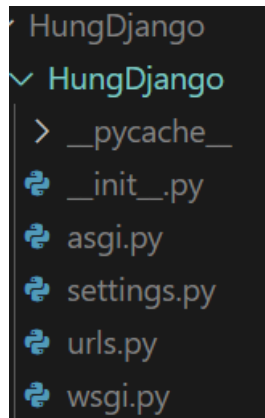
• (venv) PS V:\AppDjango\luong-quang-hung> pip install django
Collecting django
  Using cached Django-5.0.4-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.7.0 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from django)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.0.4-py3-none-any.whl (8.2 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.0-py3-none-any.whl (43 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.0.4 sqlparse-0.5.0 tzdata-2024.1

```

Hình 2.1 Tạo project

```
(venv) PS V:\AppDjango\luong-quang-hung>
(venv) PS V:\AppDjango\luong-quang-hung> django-admin startproject HungDjango
(venv) PS V:\AppDjango\luong-quang-hung>
```

Lệnh startproject sẽ tạo một thư mục có tên là HungDjango, cấu trúc bên trong thư mục sẽ gồm các file, mỗi file sẽ để làm một số công việc khác nhau như chạy server hay cài đặt cấu hình server.



Hình 2.2 Cấu trúc project

1.3. Chạy server

Sử dụng môi trường lập trình pycharm. Trong cửa sổ terminal chúng ta chạy sever bằng file manage.py với tham số là runserver.

Chúng ta chuyển đến thư mục có chứa file manage.py (hoặc dùng lệnh cd [địa chỉ thư mục] và gõ lệnh:

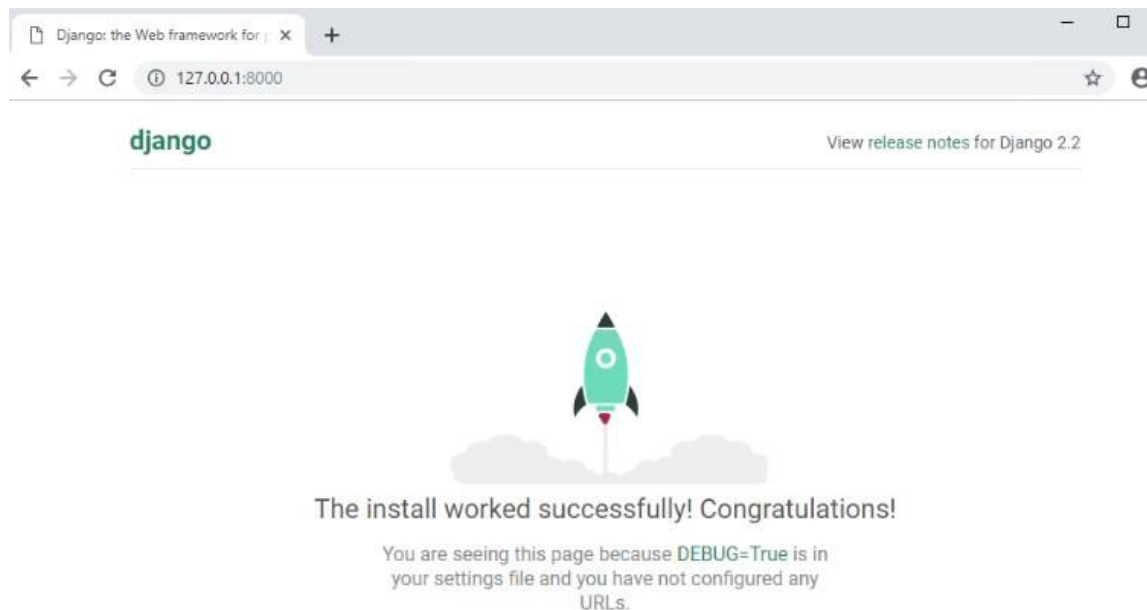
“python manage.py runserver” để chạy server

```
PS V:\AppDjango\luong-quang-hung\appWeb> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly un
pp(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
April 19, 2024 - 10:06:44
Django version 5.0.3, using setting HungDjan settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Sau đó chúng ta truy cập vào đường dẫn <http://127.0.0.1:8000/> hoặc <http://localhost:8000/> để xem chúng ta đã chạy server thành công chưa.



Hình 3.1 Chạy server

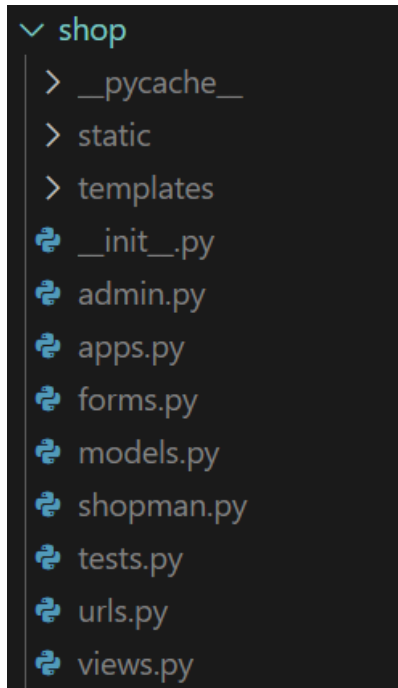
1.4. Tạo web app

Web App là một project bao gồm nhiều app, trong đó mỗi app thực hiện một công việc riêng biệt và thư mục này chứa các file chuẩn của một ứng dụng web Django.

Tạo web app bằng lệnh “*python manage.py startapp shop*”

Trong đó “*python manage.py startapp*” là lệnh để tạo web app. “**shop**” là tên app. Một thư mục với tên polls sẽ được tạo ra và có cấu trúc như sau:

myproject\HungDjango\shop



Hình 4.1 Cấu trúc App

1.5. Model

myproject\HungDjango\models.py

```
from django.db import models

class FooterInfo(models.Model):
    creation_date = models.DateField('Ngày tạo')
    company_name = models.CharField('Tên công ty', max_length=100)
    address = models.CharField('Địa chỉ', max_length=200)
    contact_name = models.CharField('Họ tên', max_length=100)
    phone_number = models.CharField('Số điện thoại', max_length=20)
    email = models.EmailField('Email')
    facebook_url = models.URLField('Facebook', blank=True)
    zalo_number = models.CharField('Zalo', max_length=20, blank=True)
    instagram_url = models.URLField('Instagram', blank=True)

    def __str__(self):
```

```

        return self.company_name

class Product_women(models.Model):
    name = models.CharField(max_length=200)
    image = models.ImageField(upload_to='products/')
    description = models.TextField()
    price = models.DecimalField(max_digits=6, decimal_places=2)
    quantity = models.IntegerField()

    def __str__(self):
        return self.name

```

Sau đó ta cấu hình trong file url.py của app shop để hiển thị view thăm dò ý kiến (list) chạy với đường dẫn là <http://localhost:8000/polls/list>.

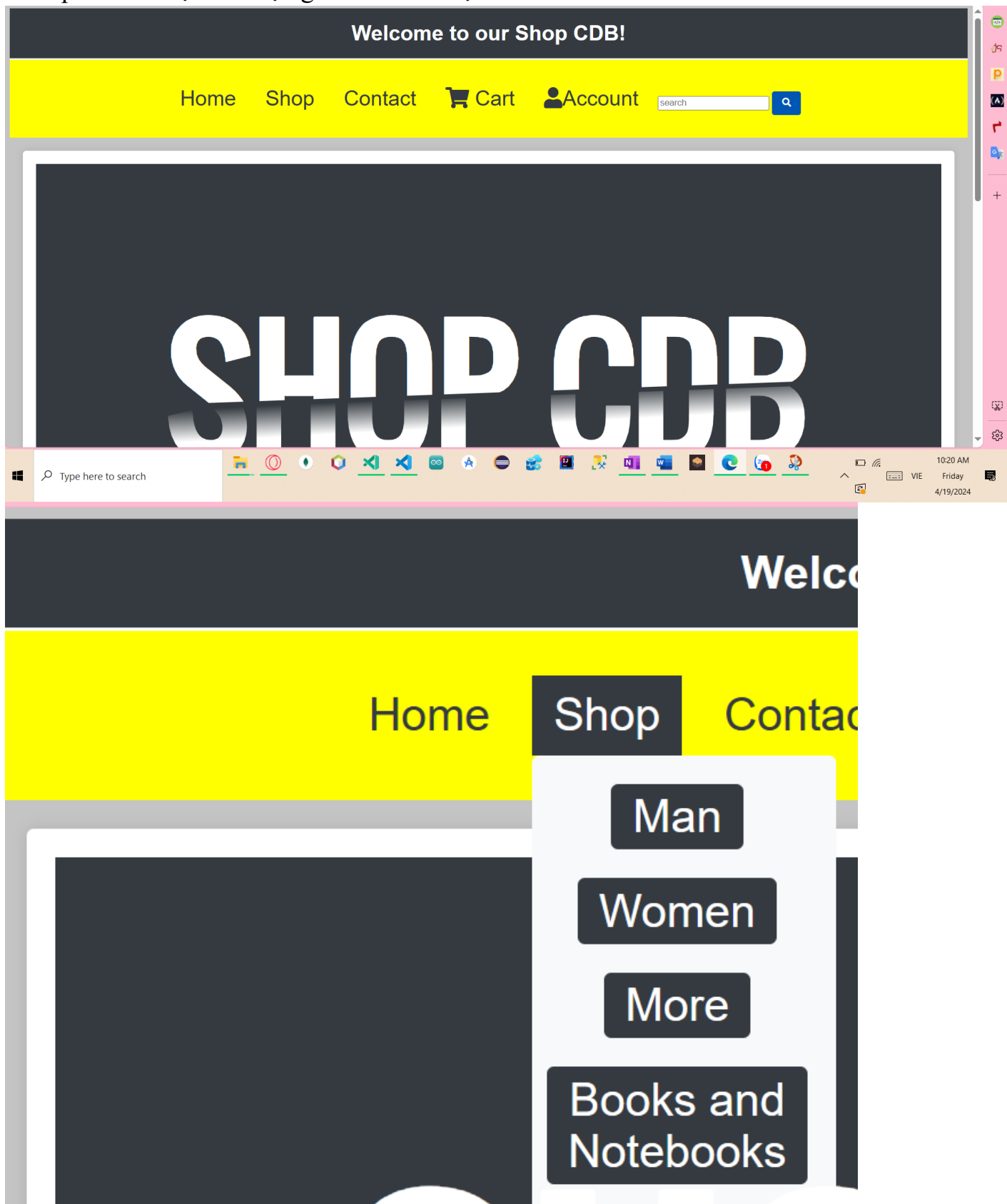
Polls/urls.py

```

from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('signup/', views.signup_view, name='signup'),
    path('signin/', views.signin_view, name='signin'),
    path('man/', views.man_view, name='man'),
    path('women/', views.women_view, name='women'),
    path('more/', views.more_view, name='more'),
    path('books-and-notbooks/', views.books_view, name='books'),
    path('product/<int:product_id>/', views.product_detail_view, name='product_detail'),
    path('search/', views.search_view, name='search'),
    path('add-to-cart/', views.add_to_cart, name='add_to_cart'),
]

```

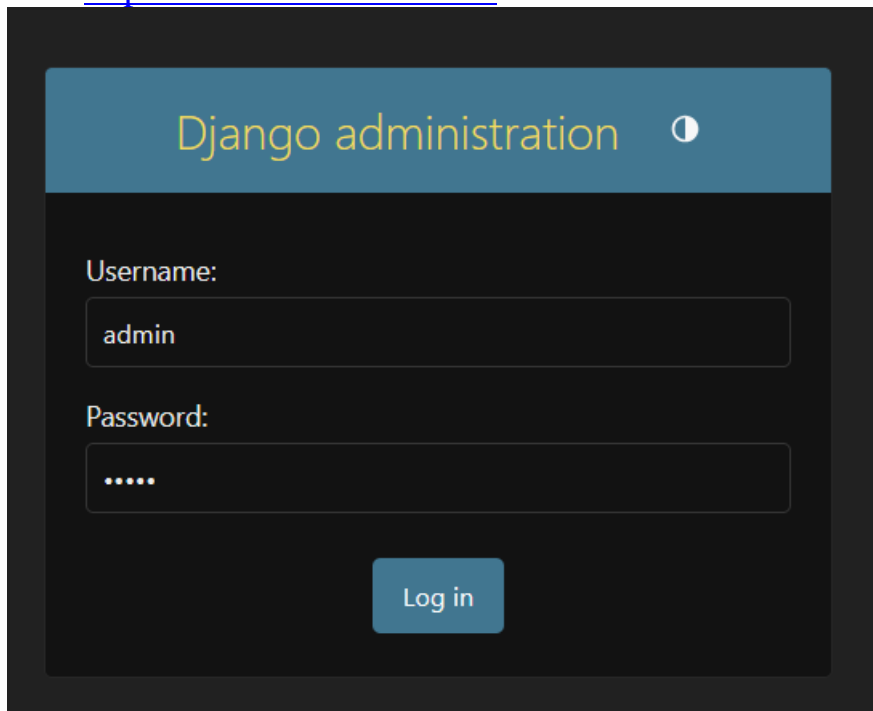
Truy cập đường dẫn <http://localhost:8000/shop> và bình chọn cho các câu hỏi.
kết quả bình chọn sẽ được gửi lên và lưu lại.



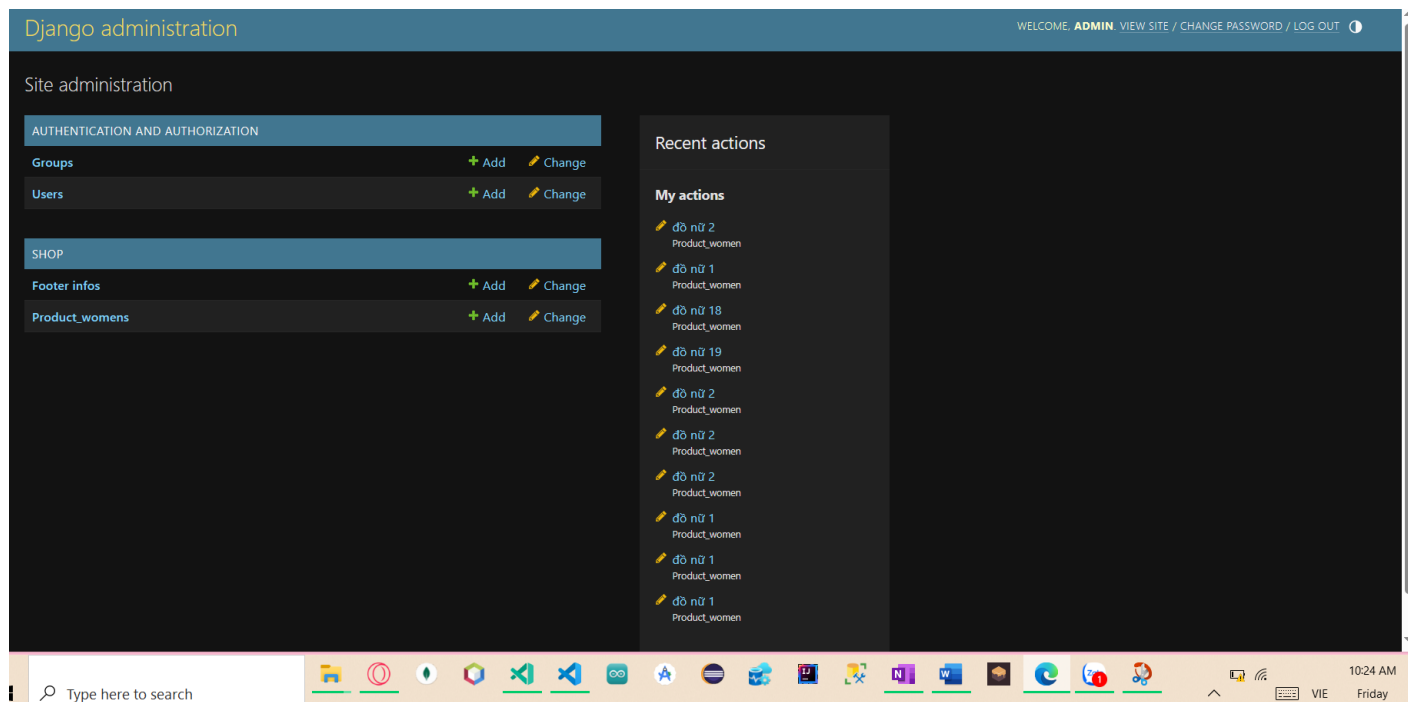
Giao diện

1.6. Hệ thống ADMIN

Khi viết một ứng dụng nào đó, chẳng hạn như website bán hàng, blog, web tin tức, diễn đàn...v.v. ngoài các trang hiển thị thông tin thì chúng ta còn phải xây dựng một trang nữa là trang admin để quản lý mọi thứ, trong đó lại bao gồm nhiều trang nhỏ hơn như thêm, sửa, xóa bài viết, cài đặt trang web. Và Django cung cấp sẵn một trang admin cho riêng chúng ta bằng cách truy cập vào đường dẫn <http://localhost:8000/admin>



Hình 6.1 Giao diện admin



Tại trang admin chúng ta có thể thao tác với 2 bảng là User và Group..lí do hai bảng đó chưa được hiển thị ở đây là vì chúng ta chưa đăng ký các bảng đó với trang admin của django nên chỉ hiển thị mặc định có hai bảng user và group.

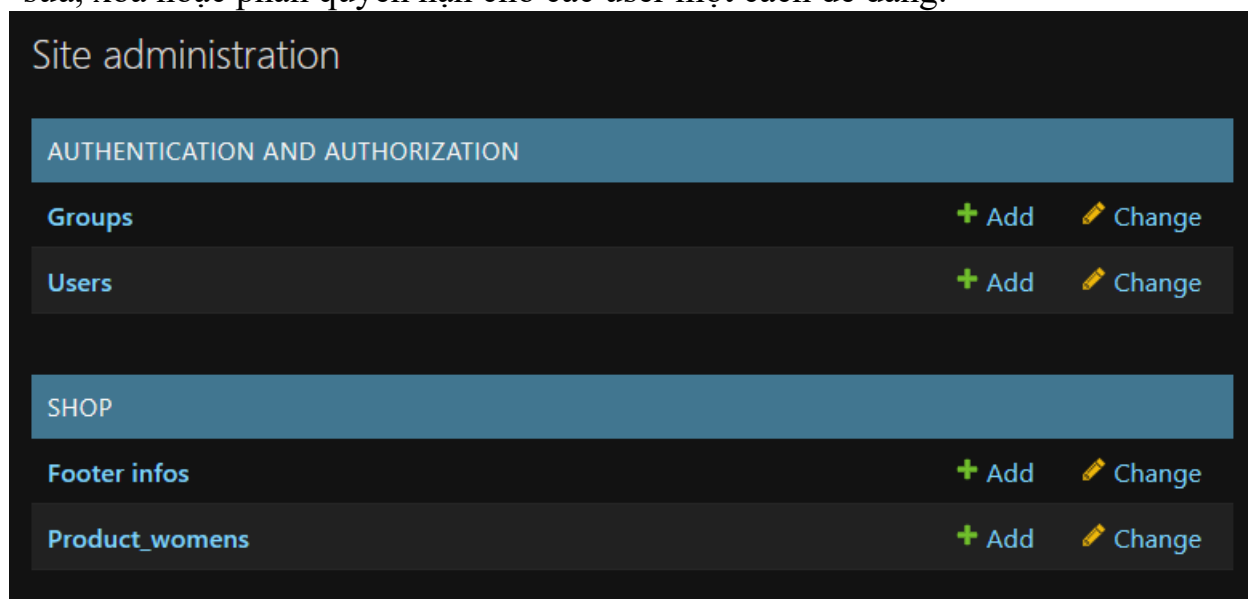
Để đăng kí các bảng (hay các mô hình) với trang admin thì chúng ta cần dùng phương thức **admin.site.register()** trong file admin.py mà Django đã tạo cho chúng ta

```
from django.contrib import admin
from .models import FooterInfo, Product_women
admin.site.register(FooterInfo)
admin.site.register(Product_women)
```

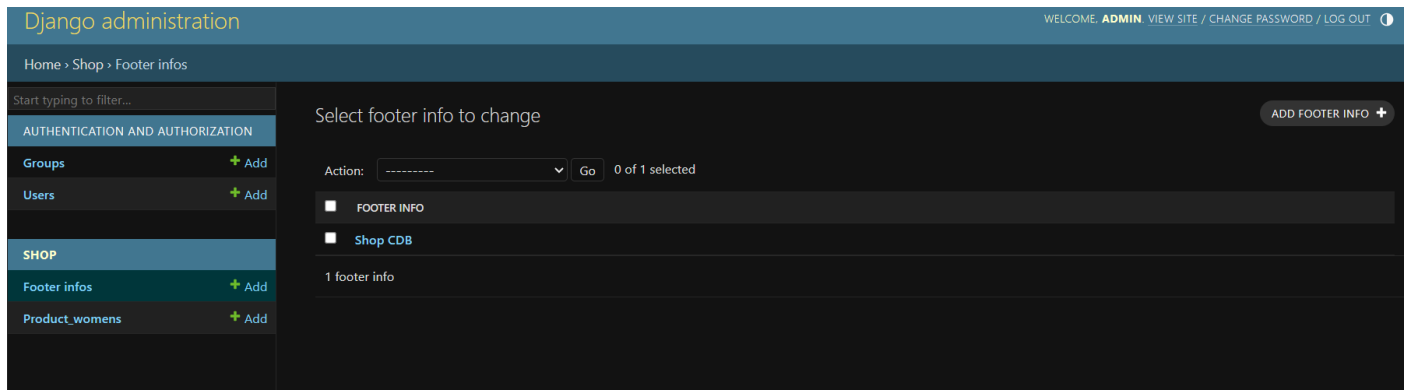
Sau khi đã đăng kí xong thì 2 bảng FooterInfo và Product sẽ hiện ra trong giao diện admin.



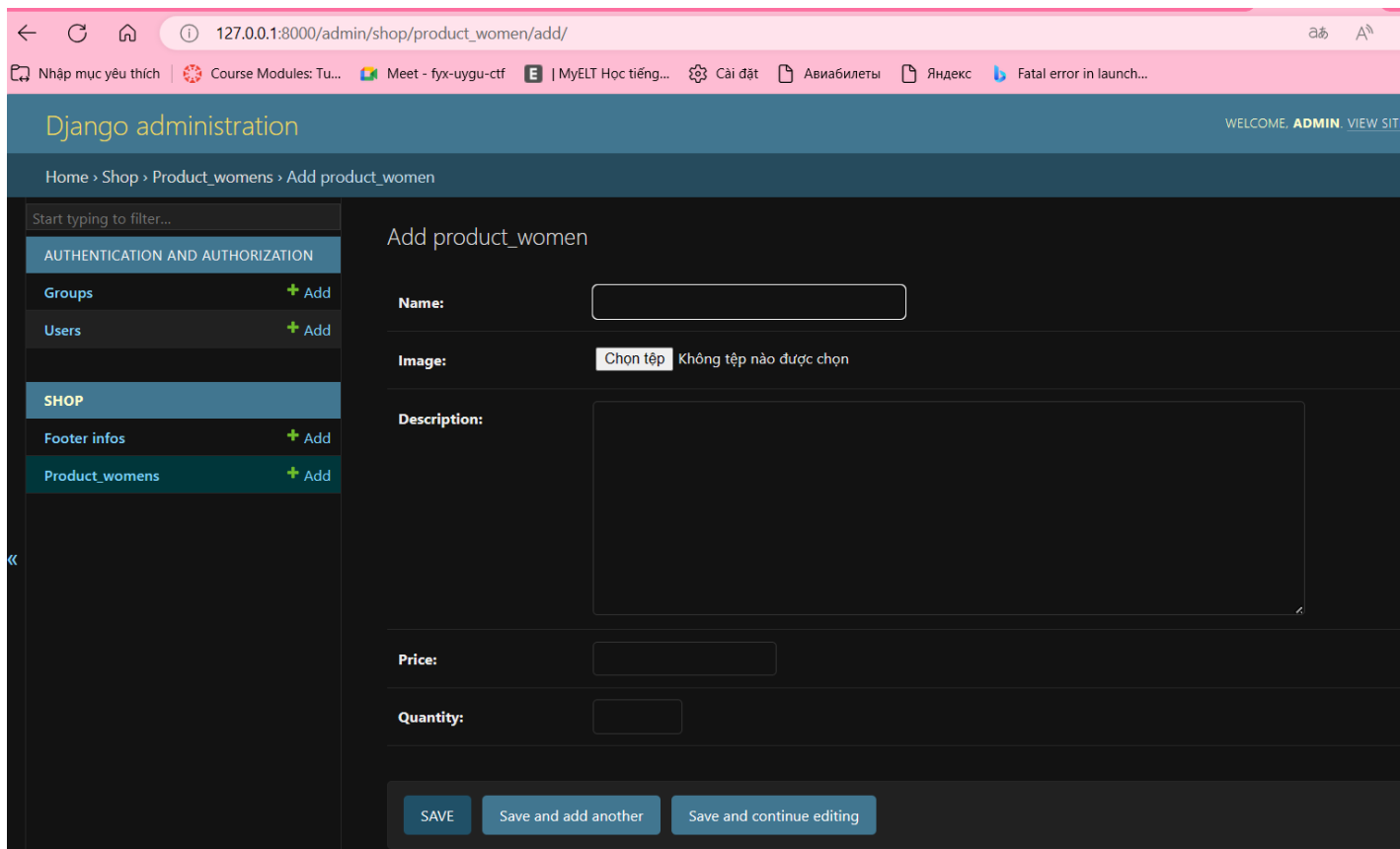
Giao diện admin mặc định của Django rất đơn giản, bạn có thể thực hiện thêm, sửa, xóa hoặc phân quyền hạn cho các user một cách dễ dàng.



Nếu chúng ta có tài khoản admin của django thì chúng ta có thể được quyền thêm, xóa người dùng, phân quyền người dùng cho các tài khoản.



Hình 6.2 Giao diện người dùng của Django



Hình 6.3 Xem cách thêm

Với quyền hạn admin chúng ta có thể cấp phép, phân quyền cho những user trong quyền hạn nhất định. Như ở đây chúng ta chỉ cho người dùng có quyền được xem câu hỏi và câu trả lời của cuộc thăm dò ý kiến, tức là người dùng này không được quyền bình chọn cũng như thêm sửa xóa .

1.7. View và Templates

a. View

Trong Django thì một View là một hàm / phương thức làm một công việc cụ thể nào đó, một view thường đi kèm với một Template.

Trong ứng dụng shop mà chúng ta đã tạo ở trên, chúng ta sẽ xây dựng các view sau.

Trong Django, một trang web được tạo ra bởi các hàm View, Django sẽ chọn View nào tùy thuộc vào URL mà chúng ta đã thiết lập. Có thể bạn đã từng thấy những đường dẫn URL nhìn rất “không đẹp mắt” như :

“ME2/Sites/dirmod.asp?sid=&type=gen&mod=Core+Pages&gid=A6CD4967199A42D9B65B1B” do website tự tạo ra, Django cho phép chúng ta tạo những đường dẫn dễ nhìn hơn. Để từ một đường dẫn URL đến một View thì Django sử dụng khái niệm URLConf, đây là một module Python của Django làm nhiệm vụ phân tích đường dẫn và chuyển đến một hàm View nhất định.

Chúng ta sẽ viết thêm một số hàm View trong file views.py

“shop/views.py”

```
from django.shortcuts import render, redirect, get_object_or_404
from .forms import SignUpForm
from .models import FooterInfo, Product_women
from django.contrib.auth import authenticate, login
from .shopman import products
from django.views.decorators.http import require_POST
from django.contrib import messages

# call home html
def home(request):
    footer_info = FooterInfo.objects.first()
    content = {'footer_info': footer_info}
    return render(request, 'home.html', content)

#sign up call html
def signup_view(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            password = form.cleaned_data.get('password1')
            # In mật khẩu ra console (chỉ nên làm điều này để debug)
            print(password)
            form.save()
            return redirect('home')
    else:
        form = SignUpForm()
    return render(request, 'signup.html', {'form': form})

# sign in call html
```

```

def signin_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('home')
        else:
            return render(request, 'signin.html', {'error': 'Username or Password
incorrect'})
    else:
        return render(request, 'signin.html')

#shop man
def man_view(request):
    footer_info = FooterInfo.objects.first()
    context = {
        'products': products,
        'footer_info': footer_info,
    }
    return render(request, 'shopCDB/man.html', context)

#shop women
def women_view(request):
    footer_info = FooterInfo.objects.first()
    shop_women = Product_women.objects.all()
    context = {
        'shop_women': shop_women,
        'footer_info': footer_info,
    }
    return render(request, 'shopCDB/women.html', context)

def more_view(request):
    return render(request, 'shopCDB/more.html')

def books_view(request):
    return render(request, 'shopCDB/books.html')

def product_detail_view(request, product_id):
    product = next((item for item in products if item["id"] == product_id), None)
    footer_info = FooterInfo.objects.first()
    context = {
        'footer_info': footer_info,
        'product': product,
    }
    return render(request, 'shopCDB/products.html', context)

def search_view(request):

```

```

query = request.GET.get('q', '')
if query:
    products = Product_women.objects.filter(name__icontains=query)
else:
    products = Product_women.objects.all()
return render(request, 'search.html', {'products': products})

@require_POST
def add_to_cart(request):
    product_id = request.POST.get('product_id')
    product = get_object_or_404(Product_women, id=product_id)
    cart = request.session.get('cart', {})
    if product_id in cart:
        cart[product_id] += 1
    else:
        cart[product_id] = 1
    request.session['cart'] = cart
    messages.add_message(request, messages.SUCCESS, 'Product added to cart successfully!')
    return redirect(request.META.get('HTTP_REFERER'))

```

Tiếp theo chúng ta tạo thêm các url dẫn đến từng view này ở trong file urls.py

“shop/urls.py”

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('signup/', views.signup_view, name='signup'),
    path('signin/', views.signin_view, name='signin'),
    path('man/', views.man_view, name='man'),
    path('women/', views.women_view, name='women'),
    path('more/', views.more_view, name='more'),
    path('books-and-notbooks/', views.books_view, name='books'),
    path('product/<int:product_id>/', views.product_detail_view, name='product_detail'),
    path('search/', views.search_view, name='search'),
    path('add-to-cart/', views.add_to_cart, name='add_to_cart'),
]
```

Khi gõ địa chỉ lên thanh URL của trình duyệt, Django sẽ đọc biến `urlpatterns` trong file `mysite.urls` vì mặc định file này được trỏ tới trong biến `ROOT_URLCONF` trong file **HungDjango/settings.py**, các đối tượng url sẽ được đọc dần dần từ trên xuống dưới cho đến khi có một đường dẫn vừa khít với URL mà bạn nhập vào.

b.Templates

Templates là một layout được thiết kế các khung web có sẵn, ta chỉ cần thêm nội dung chính của nó vào, và nhờ các template ta mới tiết kiệm thời gian trong việc phát triển website.

Trình duyệt chỉ hiểu code HTML chứ không hiểu code Python, để có thể sử dụng code Python thì Django cung cấp cho chúng ta các thẻ template, thẻ template bắt đầu và kết thúc bằng cặp kí tự `{% %}` hoặc `{{ }}`,

Các câu lệnh Python nằm trong cặp dấu `{% %}`, còn các biến thì nằm trong cặp `{{ }}`.

Ngôn ngữ Template của Django được thiết kế với mục đích chính là hỗ trợ những người đã từng làm việc với HTML, do đó nếu đã từng học HTML thì sẽ không quá khó khăn để làm quen với Template.

Ở đây trong file `base.html` (Hình 7.1) được dùng để hiển thị những phần 'tĩnh'. Một templates đơn giản để hiển thị dòng chữ “chào mừng đến với website”....

“shop/home.html”


```
<!DOCTYPE html>
<html>
{% load static %}
{% include 'style.html' %}
<body>
    <!-- Navigation -->
    {% include 'nav.html' %}
    <!-- Main Content -->
    <main>
        {% block women %}
        {% block man %}
        <section class="wrapper">
            <div class="top">Shop CDB</div>
            <div class="bottom" aria-hidden="true">Shop CDB</div>
        </section>
        <div class="image-container">
            
            
            
            
        </div>
        <h2>Đây Là Sản Phẩm Của Chúng Tôi Có!</h2>
        <p class="animated-intro">Các sản phẩm tại Shop CDB được thiết kế bởi đội ngũ nhà
thiết kế tài năng, luôn bắt kịp xu hướng thời trang mới nhất và mang đến cho bạn những
phong cách thời trang độc đáo, không thể nhầm lẫn.</p>
        {% endblock man %}
        {% endblock women %}
    </main>
    {% include 'footer.html' %}
</body>
</html>
```

Hình 7.1 cấu trúc template

File index.html được kế thừa (extends) từ home.html để hiển thị các nội dung(phần động) mà chúng ta muốn hiển thị lên cũng như tương tác với nó.

```
{% load static %}
{% include '../style.html' %}
{% include 'nav.html' %}
    {% block man %}
    <main>
    <h1>Man</h1>
```

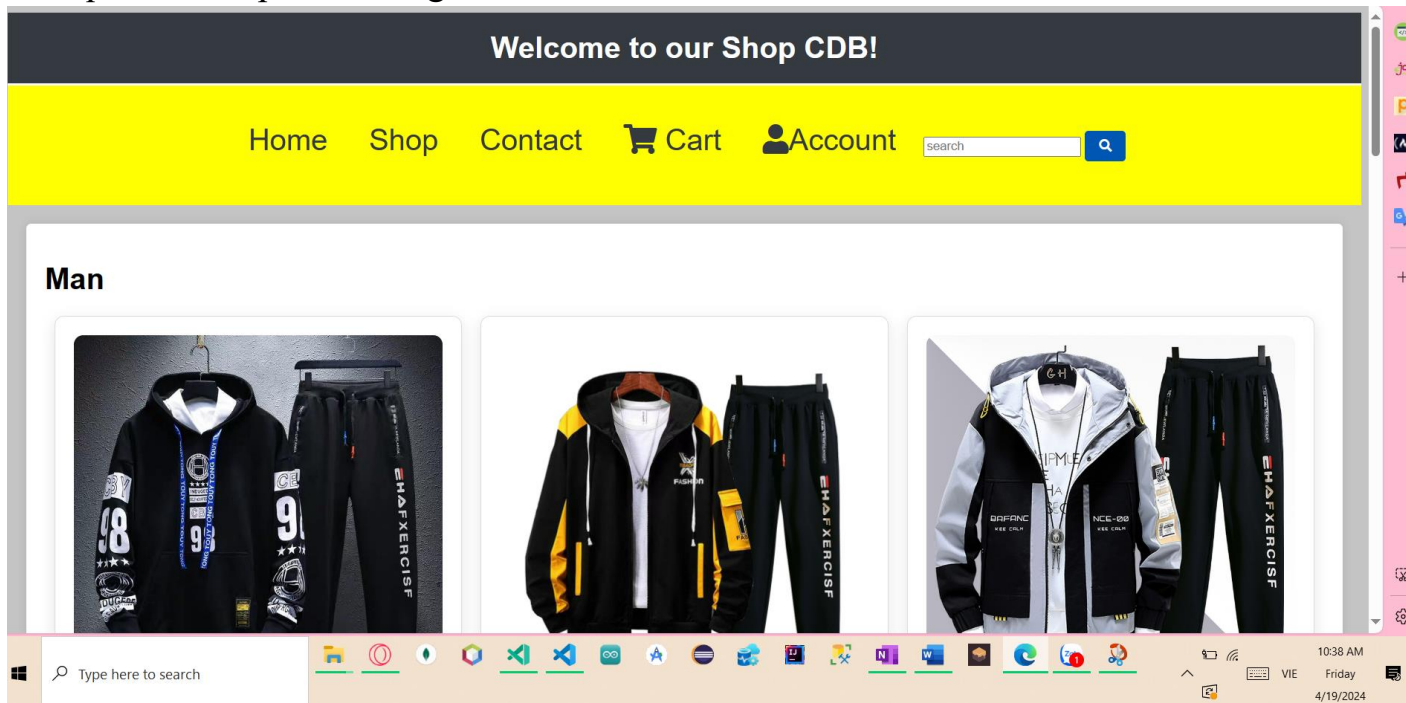
```

{% block manshow %}
<div class="products-container">
  {% for product in products %}
    <div class="product rounded-corners">
      
      <h3>{{ product.name }}</h3>
      <p>Giá: {{ product.price }}$</p>
      <button class="favorite-button">{{ product.favorite|yesno:"Favorited,Add to
favorites" }}</button>
      <a href="{% url 'product_detail' product.id %}" class="more-info-
button">Show</a>
      <button class="buy-now-button">Buy now 

```

Hình 7.2 Lệnh kế thừa

Sau đó chúng ta truy cập vào đường dẫn <http://localhost:8000/home/> để xem kết quả cho templates chúng ta vừa viết.



Nếu đã từng làm việc với các ngôn ngữ như Javascript, PHP, JSP... hay các ngôn ngữ có thể trộn chung với code HTML thì cũng nên phân biệt là Template của Django không giống các ngôn ngữ đó. Các ngôn ngữ như Javascript, PHP... là ngôn ngữ lập trình, dùng để thực hiện các công việc mang tính logic, còn HTML chỉ là ngôn ngữ đánh dấu, tức là chỉ dùng để hiển thị giao diện chứ không mang nặng phần tính toán, Template cũng vậy, đây chỉ là ngôn ngữ hỗ trợ hiển thị giao diện.

c. Đặt namespace cho URL

Khi dùng đến URL động thì lại phát sinh một vấn đề nữa, mặc định thì Django tự động tìm các file template bên trong thư mục template, vậy thì giả sử khi chúng ta có thêm nhiều ứng dụng khác ngoài shop, chẳng hạn như một ứng dụng blog, trong đó cũng có hàm view detail(), và hàm view này cũng sử dụng một template tên là detail.html, vậy thì khi đó Django sẽ gán template của ứng dụng shop vào view detail() của ứng dụng blog, như thế sẽ báo lỗi vì ứng dụng blog sẽ không có các biến giống như polls.

Để giải quyết vấn đề này, chúng ta sẽ đặt namespace cho các biến url, chúng cũng giống như một cách gộp nhóm những thứ giống nhau lại với nhau. Để đặt tên namespace cho các đối tượng url thì chúng ta chỉ cần đặt giá trị cho biến `app_name` trong file `urls.py` là được .

“shop/urls.py”

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('signup/', views.signup_view, name='signup'),
    path('signin/', views.signin_view, name='signin'),
    path('man/', views.man_view, name='man'),
    path('women/', views.women_view, name='women'),
    path('more/', views.more_view, name='more'),
    path('books-and-notbooks/', views.books_view, name='books'),
    path('product/<int:product_id>/', views.product_detail_view, name='product_detail'),
    path('search/', views.search_view, name='search'),
    path('add-to-cart/', views.add_to_cart, name='add_to_cart'),
]
```

Trong file template, ví dụ như `product_detail.html` thì chúng ta chỉ cần viết đủ tên `<namespace>:<tên biến url>` là được, ở đây url có tên là thì chỉ cần viết đầy đủ là

1.8. Upload File

b. View

Trong Django thì một View là một hàm / phương thức làm một công việc cụ thể nào đó, một view thường đi kèm với một Template.

Trong ứng dụng polls mà chúng ta đã tạo ở trên, chúng ta sẽ xây dựng các view sau:

Trong Django, một trang web được tạo ra bởi các hàm View, Django sẽ chọn View nào tùy thuộc vào URL mà chúng ta đã thiết lập. Có thể bạn đã từng thấy những đường dẫn URL nhìn rất “không đẹp mắt” như :

“ME2/Sites/dirmod.asp?sid=&type=gen&mod=Core+Pages&gid=A6CD4967199A42D9B65B1B” do website tự tạo ra, Django cho phép chúng ta tạo những đường dẫn dễ nhìn hơn. Để từ một đường dẫn URL đến một View thì Django sử dụng khái niệm URLConf, đây là một module Python của Django làm nhiệm vụ phân tích đường dẫn và chuyển đến một hàm View nhất định.

Chúng ta sẽ viết thêm một số hàm View trong file views.py

“shop/views.py”

```
from django.shortcuts import render, redirect, get_object_or_404
from .forms import SignUpForm
from .models import FooterInfo, Product_women
from django.contrib.auth import authenticate, login
from .shopman import products
from django.views.decorators.http import require_POST
from django.contrib import messages

# call home html
def home(request):
    footer_info = FooterInfo.objects.first()
    content = {'footer_info': footer_info}
    return render(request, 'home.html', content)

#sign up call html
def signup_view(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            password = form.cleaned_data.get('password1')
            # In mật khẩu ra console (chỉ nên làm điều này để debug)
            print(password)
            form.save()
            return redirect('home')
    else:
        form = SignUpForm()
    return render(request, 'signup.html', {'form': form})
```

```

# sign in call html
def signin_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('home')
        else:
            return render(request, 'signin.html', {'error': 'Username or Password
incorrect'})
    else:
        return render(request, 'signin.html')

#shop man
def man_view(request):
    footer_info = FooterInfo.objects.first()
    context = {
        'products': products,
        'footer_info': footer_info,
    }
    return render(request, 'shopCDB/man.html', context)

#shop women
def women_view(request):
    footer_info = FooterInfo.objects.first()
    shop_women = Product_women.objects.all()
    context = {
        'shop_women': shop_women,
        'footer_info': footer_info,
    }
    return render(request, 'shopCDB/women.html', context)

#shop more
def more_view(request):
    return render(request, 'shopCDB/more.html')

#shop books
def books_view(request):
    return render(request, 'shopCDB/books.html')

#show shop man
def product_detail_view(request, product_id):
    product = next((item for item in products if item["id"] == product_id), None)
    footer_info = FooterInfo.objects.first()
    context = {
        'footer_info': footer_info,
        'product': product,
    }

```

```

    }
    return render(request, 'shopCDB/products.html', context)

# search show women
def search_view(request):
    query = request.GET.get('q', '')
    if query:
        products = Product_women.objects.filter(name__icontains=query)
    else:
        products = Product_women.objects.all()
    return render(request, 'search.html', {'products': products})

@require_POST
def add_to_cart(request):
    product_id = request.POST.get('product_id')
    product = get_object_or_404(Product_women, id=product_id)
    cart = request.session.get('cart', {})
    if product_id in cart:
        cart[product_id] += 1
    else:
        cart[product_id] = 1
    request.session['cart'] = cart
    messages.add_message(request, messages.SUCCESS, 'Product added to cart successfully!')
    return redirect(request.META.get('HTTP_REFERER'))

```

Tiếp theo chúng ta tạo thêm các url dẫn đến từng view này ở trong file urls.py

“shop/urls.py”

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('signup/', views.signup_view, name='signup'),
    path('signin/', views.signin_view, name='signin'),
    path('man/', views.man_view, name='man'),
    path('women/', views.women_view, name='women'),
    path('more/', views.more_view, name='more'),
    path('books-and-notbooks/', views.books_view, name='books'),
    path('product/<int:product_id>/', views.product_detail_view, name='product_detail'),
    path('search/', views.search_view, name='search'),
    path('add-to-cart/', views.add_to_cart, name='add_to_cart'),
]
```

Khi gõ địa chỉ lên thanh URL của trình duyệt, Django sẽ đọc biến `urlpatterns` trong file `mysite.urls` vì mặc định file này được trỏ tới trong biến `ROOT_URLCONF` trong file **HungDjango/settings.py**, các đối tượng url sẽ được đọc dần dần từ trên xuống dưới cho đến khi có một đường dẫn vừa khít với URL mà bạn nhập vào.

b.Templates

Templates là một layout được thiết kế các khung web có sẵn, ta chỉ cần thêm nội dung chính của nó vào, và nhờ các template ta mới tiết kiệm thời gian trong việc phát triển website.

Trình duyệt chỉ hiểu code HTML chứ không hiểu code Python, để có thể sử dụng code Python thì Django cung cấp cho chúng ta các thẻ template, thẻ template bắt đầu và kết thúc bằng cặp kí tự `{% %}` hoặc `{{ }}`,

Các câu lệnh Python nằm trong cặp dấu `{% %}`, còn các biến thì nằm trong cặp `{{ }}`.

Ngôn ngữ Template của Django được thiết kế với mục đích chính là hỗ trợ những người đã từng làm việc với HTML, do đó nếu đã từng học HTML thì sẽ không quá khó khăn để làm quen với Template.

Ở đây trong file `base.html` (Hình 7.1) được dùng để hiển thị những phần 'tĩnh'. Một templates đơn giản để hiển thị dòng chữ “chào mừng đến với website”....

“shop/ templates”

```
<!DOCTYPE html>
<html>
{% load static %}
{% include 'style.html' %}
<body>
    <!-- Navigation -->
    {% include 'nav.html' %}
    <!-- Main Content -->
    <main>
        {% block women %}
        {% block man %}
        <section class="wrapper">
            <div class="top">Shop CDB</div>
            <div class="bottom" aria-hidden="true">Shop CDB</div>
        </section>
        <div class="image-container">
            
            
            
            
        </div>
        <h2>Đây Là Sản Phẩm Của Chúng Tôi Có!</h2>
        <p class="animated-intro">Các sản phẩm tại Shop CDB được thiết kế bởi đội ngũ nhà
thiết kế tài năng, luôn bắt kịp xu hướng thời trang mới nhất và mang đến cho bạn những
phong cách thời trang độc đáo, không thể nhầm lẫn.</p>
        {% endblock man %}
        {% endblock women %}
    </main>
    {% include 'footer.html' %}
</body>
</html>
```

- ▼ shop
 - > __pycache__
 - > static
 - ▼ templates
 - ▼ shopCDB
 - <> books.html
 - <> man.html
 - <> more.html
 - <> products.html
 - <> women.html
 - <> footer.html
 - <> home.html
 - <> javascript.html
 - <> nav.html
 - <> search.html
 - <> signin.html
 - <> signup.html
 - <> style.html

```
<!-- Footer -->
<footer style="background-color: #343a40; text-align: center; padding: 20px; color: white; width: 100%; display: flex; justify-content: space-around;">
  <div style="margin-bottom: 20px;">
    <h3> Website Information </h3>
    <p>Ngày tạo: {{ footer_info.creation_date }}</p>
    <p>Company: {{ footer_info.company_name }}</p>
    <p>Địa chỉ: {{ footer_info.address }}</p>
  </div>
  <div style="margin-bottom: 20px;">
    <h3> Contact </h3>
    <p>Name: {{ footer_info.contact_name }}</p>
    <p>Number Phone: {{ footer_info.phone_number }}</p>
    <p>Email: {{ footer_info.email }}</p>
  </div>
  <div>
    <h3> Social network </h3>
    <p>Facebook: <a href="{{ footer_info.facebook_url }}" style="color: white;">Quang Hung</a></p>
    <p>Zalo: {{ footer_info.zalo_number }}</p>
    <p>Instagram: <a href="{{ footer_info.instagram_url }}" style="color: white;">Hung2003</a></p>
  </div>
```

```
</footer>
```

```
<header>
```

```
  <h1>Welcome to our Shop CDB!</h1>
```

```
</header>
```

```
<nav>
```

```
  <ul>
```

```
    <li>
```

```
      <a href="{% url 'home' %}">Home</a>
```

```
    </li>
```

```
    <li>
```

```
      <a href="{% url 'home' %}">Shop</a>
```

```
      <ul>
```

```
        <li>
```

```
          <a href="{% url 'man' %}" class="nav-item" >Man</a>
```

```
        </li>
```

```
        <li>
```

```
          <a href="{% url 'women' %}" class="nav-item" >Women</a>
```

```
        </li>
```

```
        <li>
```

```
          <a href="{% url 'more' %}" class="nav-item" >More</a>
```

```
        </li>
```

```
        <li>
```

```
          <a href="{% url 'books' %}" class="nav-item">Books and Notebooks</a>
```

```
        </li>
```

```
      </ul>
```

```
    </li>
```

```
    <li><a href="#">Contact</a></li>
```

```
    <li>
```

```
      <a href="#">
```

```
        <i class="fas fa-shopping-cart"></i>
```

```
        Cart
```

```
      </a>
```

```
    </li>
```

```
  <li>
```

```
    <a href="{% url 'home' %}"><i class="fas fa-user"></i>Account</a>
```

```
    <ul>
```

```
      <li>
```

```
        <a href="{% url 'signin' %}" class="nav-item">Sign In</a>
```

```
      </li>
```

```
      <li>
```

```
        <a href="{% url 'signup' %}" class="nav-item">Sign Up</a>
```

```
      </li>
```

```
    </ul>
```

```
  </li>
```

```
  <li>
```

```
    <form action="{% url 'search' %}" method="get" class="search-form">
```

```

        <input type="text" name="q" placeholder="search" required />
        <button type="submit" class="search-button">
            <i class="fas fa-search"></i>
        </button>
    </form>
</li>
</ul>
</nav>


{% load static %}
<head>
    <title>Shop</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <link rel="stylesheet" href="{% static '/style/home.css' %}">
    <link rel="stylesheet" href="{% static '/style/shopman.css' %}">
    <link rel="stylesheet" href="{% static '/style/showman.css' %}">
</head>

```

Hình 7.1 cấu trúc template

File man.html được kế thừa (extends) từ home.html để hiển thị các nội dung(phần động) mà chúng ta muốn hiển thị lên cũng như tương tác với nó.

```

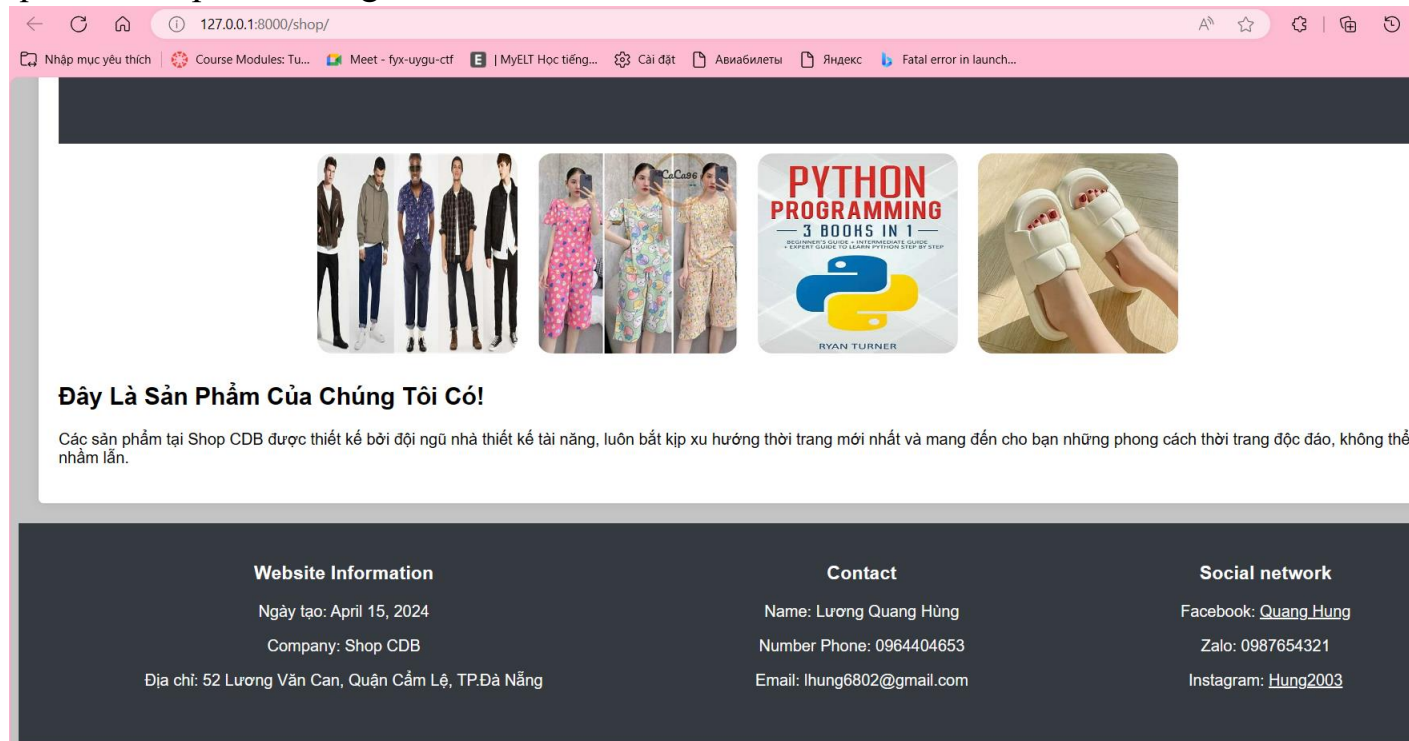
{% load static %}
{% include '../style.html' %}
{% include 'nav.html' %}
    {% block man %}
        <main>
            <h1>Man</h1>
            {% block manshow %}
                <div class="products-container">
                    {% for product in products %}
                        <div class="product rounded-corners">
                            
                            <h3>{{ product.name }}</h3>
                            <p>Giá: {{ product.price }}$</p>
                            <button class="favorite-button">{{ product.favorite|yesno:"Favorited,Add to
favorites" }}</button>
                            <a href="{% url 'product_detail' product.id %}" class="more-info-
button">Show</a>
                            <button class="buy-now-button">Buy now 

```




Hình 7.2 Lệnh kế thừa

Sau đó chúng ta truy cập vào đường dẫn <http://localhost:8000/shop/> để xem kết quả cho templates chúng ta vừa viết.



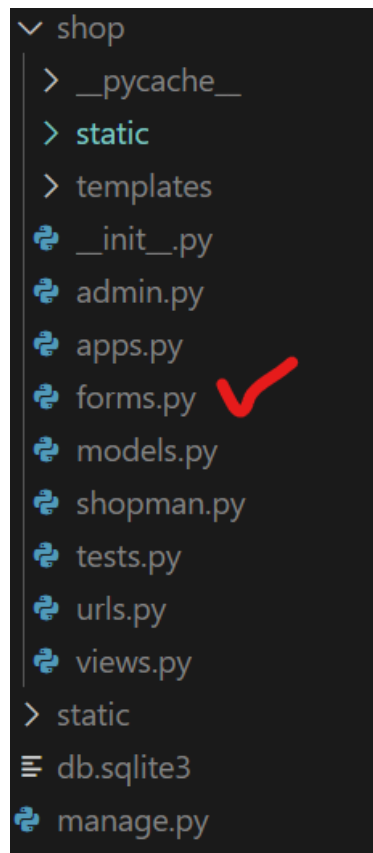
1.9. Form trong Django

c. file form sử dụng model

Trong HTML thì form là một tập các thẻ element nằm giữa cặp thẻ `<form>...</form>` cho phép người dùng thực hiện một công việc nào đó, sau đó gửi dữ liệu lên server rồi trả về câu trả lời. Hầu hết HTML chỉ hỗ trợ các thẻ element cơ bản như text, checkbox... muốn hiển thị các element cao cấp hơn như DateTimePicker (chọn ngày tháng), Slider (thanh trượt)... phải dùng đến các ngôn ngữ hỗ trợ thêm khác như Javascript, CSS... Hoặc có thể dùng lớp Form có sẵn của Django, Django cung cấp lớp Form hỗ trợ tạo form một cách nhanh chóng và dễ dàng.

Dùng form để nhập dữ liệu sau đó đổ dữ liệu vào trong model thế nên dùng model form sẽ tiết kiệm thời gian hơn.ở đây chúng ta sẽ làm riêng 1 project để làm về form , cụ thể ở đây chúng ta sẽ thiết kế form để nhập dữ liệu sau đó gửi lên server và lưu lại.

Ở đây chúng ta tạo mới một project có tên **djangoform** để ví dụ về phần model form,với chức năng đơn giản nhập dữ liệu (đăng bài viết hoặc tin tức) gửi lên server và lưu lại.. tạo form.py



Hình 9.1 Tạo form

“HungDjango/settings.py”

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'shop',  
    'introduce',  
]
```

Trong mỗi app có những phần gọi là router, nó sẽ định tuyến từng đường link ở

trong từng app nên ta sẽ tạo một file.py trong app news có tên là “urls” và import vào file views.

“HungDjango/urls.py”

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('signup/', views.signup_view, name='signup'),
    path('signin/', views.signin_view, name='signin'),
    path('man/', views.man_view, name='man'),
    path('women/', views.women_view, name='women'),
    path('more/', views.more_view, name='more'),
    path('books-and-notbooks/', views.books_view, name='books'),
    path('product/<int:product_id>/', views.product_detail_view,
name='product_detail'),
    path('search/', views.search_view, name='search'),
    path('add-to-cart/', views.add_to_cart, name='add_to_cart'),
]
```

Hình 9.2

forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class SignUpForm(UserCreationForm):
    first_name = forms.CharField(max_length=30, required=True)
    last_name = forms.CharField(max_length=30, required=True)
    phone_number = forms.CharField(max_length=15, required=True)
    address = forms.CharField(max_length=100, required=True)
    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'phone_number',
'address', 'password1', 'password2',)
```

```
from django.shortcuts import render, redirect, get_object_or_404
from .forms import SignUpForm
from .models import FooterInfo, Product_women
from django.contrib.auth import authenticate, login
from .shopman import products
from django.views.decorators.http import require_POST
from django.contrib import messages
```

```

# call home html
def home(request):
    footer_info = FooterInfo.objects.first()
    content = {'footer_info': footer_info}
    return render(request, 'home.html', content)

#sign up call html
def signup_view(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            password = form.cleaned_data.get('password1')
            # In mật khẩu ra console (chỉ nên làm điều này để debug)
            print(password)
            form.save()
            return redirect('home')
        else:
            form = SignUpForm()
    return render(request, 'signup.html', {'form': form})

# sign in call html
def signin_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('home')
        else:
            return render(request, 'signin.html', {'error': 'Username or Password incorrect'})
    else:
        return render(request, 'signin.html')

#shop man
def man_view(request):
    footer_info = FooterInfo.objects.first()
    context = {
        'products': products,
        'footer_info': footer_info,
    }
    return render(request, 'shopCDB/man.html', context)

#shop women
def women_view(request):
    footer_info = FooterInfo.objects.first()
    shop_women = Product_women.objects.all()

```

```

        context = {
            'shop_women': shop_women,
            'footer_info': footer_info,
        }
        return render(request, 'shopCDB/women.html', context)

#shop more
def more_view(request):
    return render(request, 'shopCDB/more.html')

#shop books
def books_view(request):
    return render(request, 'shopCDB/books.html')

#show shop man
def product_detail_view(request, product_id):
    product = next((item for item in products if item["id"] == product_id),
None)
    footer_info = FooterInfo.objects.first()
    context = {
        'footer_info': footer_info,
        'product': product,
    }
    return render(request, 'shopCDB/products.html', context)

# search show women
def search_view(request):
    query = request.GET.get('q', '')
    if query:
        products = Product_women.objects.filter(name__icontains=query)
    else:
        products = Product_women.objects.all()
    return render(request, 'search.html', {'products': products})

@require_POST
def add_to_cart(request):
    product_id = request.POST.get('product_id')
    product = get_object_or_404(Product_women, id=product_id)
    cart = request.session.get('cart', {})
    if product_id in cart:
        cart[product_id] += 1
    else:
        cart[product_id] = 1
    request.session['cart'] = cart

```

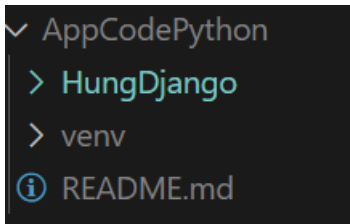
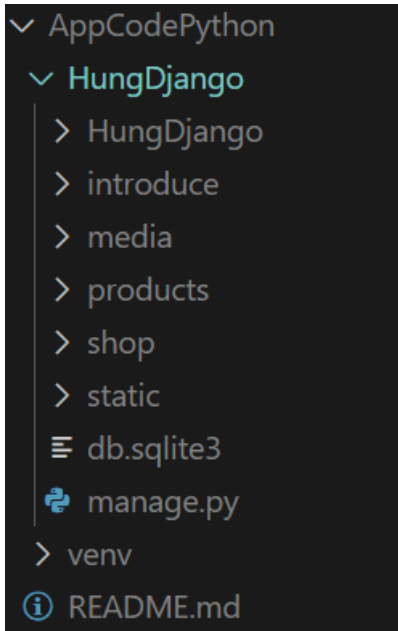
```
messages.add_message(request, messages.SUCCESS, 'Product added to cart successfully!')
return redirect(request.META.get('HTTP_REFERER'))
```

Hình 9.3 Tạo class.





Sau đó chúng ta tạo một file có tên là **forms.py** để nhập dữ liệu cho class vừa tạo và post nó lên và lưu vào trong database. Thêm các trường để hiển thị ra bên ngoài forms (tiêu đề bài viết, nội dung, thời gian khởi tạo).











TEST PROJECT

Bảng Code

- ✓ AppCodePython
- ✓ HungDjango
 - ✓ HungDjango
 - > __pycache__
 - 🔗 __init__.py
 - 🔗 asgi.py
 - 🔗 settings.py
 - 🔗 urls.py
 - 🔗 wsgi.py
 - > introduce
 - > media
 - > products
 - > shop
 - > static
 - ≡ db.sqlite3
 - 🔗 manage.py
 - > venv

```
✓ AppCodePython
  ✓ HungDjango
    > HungDjango
    > introduce
    ✓ media\products
       do-nu-01_hH5FwmG.jpg
       do-nu-01.jpg
       do-nu-02_0Sp96Mx.jpg
       do-nu-02.jpg
      ≡ do-nu-03.jfif
      ≡ do-nu-04.jfif
      ≡ do-nu-05.jfif
      ≡ do-nu-06.jfif
      ≡ do-nu-07.jfif
      ≡ do-nu-08.jfif
      ≡ do-nu-09.jfif
```

```
✓ AppCodePython
  ✓ HungDjango
    ✓ shop
      > __pycache__
      > static
      > templates
       __init__.py
       admin.py
       apps.py
       forms.py
       models.py
       shopman.py
       tests.py
       urls.py
       views.py
      > static
      ≡ db.sqlite3
       manage.py
```

```
✓ HungDjango
  ✓ shop
    ✓ templates
      ✓ shopCDB
        <> books.html
        <> man.html
        <> more.html
        <> products.html
        <> women.html
      <> footer.html
      <> home.html
      <> javascript.html
      <> nav.html
      <> search.html
      <> signin.html
      <> signup.html
      <> style.html
```

admin.py

```
from django.contrib import admin
from .models import FooterInfo, Product_women
admin.site.register(FooterInfo)
admin.site.register(Product_women)
```

forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User
class SignUpForm(UserCreationForm):
    first_name = forms.CharField(max_length=30, required=True)
    last_name = forms.CharField(max_length=30, required=True)
    phone_number = forms.CharField(max_length=15, required=True)
    address = forms.CharField(max_length=100, required=True)
    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'phone_number', 'address',
'password1', 'password2',)
```

models.py

```
from django.db import models

class FooterInfo(models.Model):
    creation_date = models.DateField('Ngày tạo')
    company_name = models.CharField('Tên công ty', max_length=100)
    address = models.CharField('Địa chỉ', max_length=200)
    contact_name = models.CharField('Họ tên', max_length=100)
    phone_number = models.CharField('Số điện thoại', max_length=20)
    email = models.EmailField('Email')
    facebook_url = models.URLField('Facebook', blank=True)
    zalo_number = models.CharField('Zalo', max_length=20, blank=True)
    instagram_url = models.URLField('Instagram', blank=True)
    def __str__(self):
        return self.company_name

class Product_women(models.Model):
    name = models.CharField(max_length=200)
    image = models.ImageField(upload_to='products/')
    description = models.TextField()
    price = models.DecimalField(max_digits=6, decimal_places=2)
    quantity = models.IntegerField()
    def __str__(self):
        return self.name
```

shopman.py

```
#shop man
products = [
    {
        'id': 1,
        'name': 'shirt man red',
        'image': 'images/do-nam-01.jpg',
        'price': '20',
        'favorite': False,
        'description': 'Áo sơ mi nam màu đỏ, chất liệu cotton mềm mại, phong cách trẻ trung.'
    },
    {
        'id': 2,
        'name': 'shirt man yellow SC',
        'image': 'images/do-nam-02.jpg',
        'price': '30',
        'favorite': True,
        'description': 'Áo sơ mi nam màu xanh dương, thiết kế hiện đại, phù hợp mọi dịp.'
    },
    {
        'id': 3,
        'name': 'shirt man yellow',
        'image': 'images/do-nam-03.jpg',
```

```
        'price': '45',
        'favorite': True,
        'description': 'Áo sơ mi nam màu vàng, tạo điểm nhấn nổi bật, chất liệu thoáng mát.'
    },
    {
        'id': 4,
        'name': 'shirt man gray',
        'image': 'images/do-nam-04.jpg',
        'price': '60',
        'favorite': False,
        'description': 'Áo sơ mi nam màu xám, kiểu dáng lịch sự, dễ phối đồ.'
    },
    {
        'id': 5,
        'name': 'shirt man black',
        'image': 'images/do-nam-05.jpg',
        'price': '60',
        'favorite': False,
        'description': 'Áo sơ mi nam màu đen, chất liệu cao cấp, mang lại vẻ đẳng cấp.'
    },
    {
        'id': 6,
        'name': 'shirt man black',
        'image': 'images/do-nam-06.jpg',
        'price': '60',
        'favorite': False,
        'description': 'Áo sơ mi nam màu đen, phong cách cổ điển, phù hợp mọi lứa tuổi.'
    },
    {
        'id': 7,
        'name': 'shirt man black',
        'image': 'images/do-nam-07.jpg',
        'price': '60',
        'favorite': False,
        'description': 'Áo sơ mi nam màu đen, thiết kế đơn giản, thích hợp cho môi trường công sở.'
    },
    {
        'id': 8,
        'name': 'shirt man black',
        'image': 'images/do-nam-08.jpg',
        'price': '60',
        'favorite': True,
```

```

        'description': 'Áo sơ mi nam màu đen, chất liệu bền bỉ, kiểu dáng thời trang.'
    },
    {
        'id': 9,
        'name': 'shirt man black',
        'image': 'images/do-nam-09.jpg',
        'price': '60',
        'favorite': False,
        'description': 'Áo sơ mi nam màu đen, phù hợp với các buổi tiệc tối hoặc sự kiện
    },
    {
        'id': 10,
        'name': 'shirt man black',
        'image': 'images/do-nam-10.jpg',
        'price': '60',
        'favorite': False,
        'description': 'Áo sơ mi nam màu đen, mang lại cảm giác thoải mái, phong cách đa
dạng.'
    },
]

```

urls.py

```

from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('signup/', views.signup_view, name='signup'),
    path('signin/', views.signin_view, name='signin'),
    path('man/', views.man_view, name='man'),
    path('women/', views.women_view, name='women'),
    path('more/', views.more_view, name='more'),
    path('books-and-notbooks/', views.books_view, name='books'),
    path('product/<int:product_id>/', views.product_detail_view, name='product_detail'),
    path('search/', views.search_view, name='search'),
    path('add-to-cart/', views.add_to_cart, name='add_to_cart'),
]

```

views.py

```

from django.shortcuts import render, redirect, get_object_or_404
from .forms import SignUpForm
from .models import FooterInfo, Product_women
from django.contrib.auth import authenticate, login
from .shopman import products
from django.views.decorators.http import require_POST
from django.contrib import messages

# call home html
def home(request):
    footer_info = FooterInfo.objects.first()

```

```

        content = {'footer_info': footer_info}
        return render(request, 'home.html', content)

#sign up call html
def signup_view(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            password = form.cleaned_data.get('password1')
            # In mật khẩu ra console (chỉ nên làm điều này để debug)
            print(password)
            form.save()
            return redirect('home')
        else:
            form = SignUpForm()
    return render(request, 'signup.html', {'form': form})

# sign in call html
def signin_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('home')
        else:
            return render(request, 'signin.html', {'error': 'Username or Password
incorrect'})
    else:
        return render(request, 'signin.html')

#shop man
def man_view(request):
    footer_info = FooterInfo.objects.first()
    context = {
        'products': products,
        'footer_info': footer_info,
    }
    return render(request, 'shopCDB/man.html', context)

#shop women
def women_view(request):
    footer_info = FooterInfo.objects.first()

```

```

shop_women = Product_women.objects.all()
context = {
    'shop_women': shop_women,
    'footer_info': footer_info,
}
return render(request, 'shopCDB/women.html', context)

#shop more
def more_view(request):
    return render(request, 'shopCDB/more.html')

#shop books
def books_view(request):
    return render(request, 'shopCDB/books.html')

#show shop man
def product_detail_view(request, product_id):
    product = next((item for item in products if item["id"] == product_id), None)
    footer_info = FooterInfo.objects.first()
    context = {
        'footer_info': footer_info,
        'product': product,
    }
    return render(request, 'shopCDB/products.html', context)

# search show women
def search_view(request):
    query = request.GET.get('q', '')
    if query:
        products = Product_women.objects.filter(name__icontains=query)
    else:
        products = Product_women.objects.all()
    return render(request, 'search.html', {'products': products})

@require_POST
def add_to_cart(request):
    product_id = request.POST.get('product_id')
    product = get_object_or_404(Product_women, id=product_id)
    cart = request.session.get('cart', {})
    if product_id in cart:
        cart[product_id] += 1
    else:
        cart[product_id] = 1
    request.session['cart'] = cart
    messages.add_message(request, messages.SUCCESS, 'Product added to cart successfully!')
    return redirect(request.META.get('HTTP_REFERER'))

```

home.html

<!DOCTYPE html>


```

<html>
{% load static %}
{% include 'style.html' %}
<body>
    <!-- Navigation -->
    {% include 'nav.html' %}
    <!-- Main Content -->
    <main>
        {% block women %}
        {% block man %}
        <section class="wrapper">
            <div class="top">Shop CDB</div>
            <div class="bottom" aria-hidden="true">Shop CDB</div>
        </section>
        <div class="image-container">
            
            
            
            
        </div>
        <h2>Đây Là Sản Phẩm Của Chúng Tôi Có!</h2>
        <p class="animated-intro">Các sản phẩm tại Shop CDB được thiết kế bởi đội ngũ nh
thiết kế tài năng, luôn bắt kịp xu hướng thời trang mới nhất và mang đến cho bạn những
phong cách thời trang độc đáo, không thể nhầm lẫn.</p>
        {% endblock man %}
        {% endblock women %}
    </main>
    {% include 'footer.html' %}
</body>
</html>

```

nav.html

```

<header>
    <h1>Welcome to our Shop CDB!</h1>
</header>
<nav>
    <ul>
        <li>
            <a href="{% url 'home' %}">Home</a>
        </li>
        <li>
            <a href="{% url 'home' %}">Shop</a>

```

```

        <ul>
            <li>
                <a href="{% url 'man' %}" class="nav-item" >Man</a>
            </li>
            <li>
                <a href="{% url 'women' %}" class="nav-item" >Women</a>
            </li>
            <li>
                <a href="{% url 'more' %}" class="nav-item" >More</a>
            </li>
            <li>
                <a href="{% url 'books' %}" class="nav-item">Books and Notebooks</a>
            </li>
        </ul>
    </li>
    <li><a href="#">Contact</a></li>
    <li>
        <a href="#">
            <i class="fas fa-shopping-cart"></i>
            Cart
        </a>
    </li>
    <li>
        <a href="{% url 'home' %}"><i class="fas fa-user"></i>Account</a>
        <ul>
            <li>
                <a href="{% url 'signin' %}" class="nav-item">Sign In</a>
            </li>
            <li>
                <a href="{% url 'signup' %}" class="nav-item">Sign Up</a>
            </li>
        </ul>
    </li>
    <li>
        <form action="{% url 'search' %}" method="get" class="search-form">
            <input type="text" name="q" placeholder="search" required />
            <button type="submit" class="search-button">
                <i class="fas fa-search"></i>
            </button>
        </form>
    </li>
</ul>
</nav>

```

search.html

```

{% include 'style.html' %}
{% include 'nav.html' %}
{% load static %}
{% block search %}

```

```

<h2>Kết quả tìm kiếm</h2>
<div class="products-container">
  {% for product in products %}
    <div class="product">
      <!-- Sử dụng 'product.image.url' để lấy URL của hình ảnh -->
      
      <h3>{{ product.name }}</h3>
      <p>Giá: {{ product.price }}$</p>
      <p>{{ product.description }}</p>
      <a href="{% url 'product_detail' product.id %}">Xem chi tiết</a>
    </div>
  {% empty %}
    <p>Không tìm thấy sản phẩm nào.</p>
  {% endfor %}
</div>
{% endblock %}
{% include 'footer.html' %}

```

footer.html

```

<!-- Footer -->
<footer style="background-color: #343a40; text-align: center; padding: 20px; color: white; width: 100%; display: flex; justify-content: space-around;">
  <div style="margin-bottom: 20px;">
    <h3>Website Information</h3>
    <p>Ngày tạo: {{ footer_info.creation_date }}</p>
    <p>Company: {{ footer_info.company_name }}</p>
    <p>Địa chỉ: {{ footer_info.address }}</p>
  </div>
  <div style="margin-bottom: 20px;">
    <h3>Contact</h3>
    <p>Name: {{ footer_info.contact_name }}</p>
    <p>Number Phone: {{ footer_info.phone_number }}</p>
    <p>Email: {{ footer_info.email }}</p>
  </div>
  <div>
    <h3>Social network</h3>
    <p>Facebook: <a href="{{ footer_info.facebook_url }}" style="color: white;">Quang Hung</a></p>
    <p>Zalo: {{ footer_info.zalo_number }}</p>
    <p>Instagram: <a href="{{ footer_info.instagram_url }}" style="color: white;">Hung2003</a></p>
  </div>
</footer>

```

style.html

```
{% load static %}
<head>
    <title>Shop</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <link rel="stylesheet" href="{% static '/style/home.css' %}">
    <link rel="stylesheet" href="{% static '/style/shopman.css' %}">
    <link rel="stylesheet" href="{% static '/style/showman.css' %}">
</head>
```

signin.html

```
<!DOCTYPE html>
<html>
<head>
    {% load static %}
    <title>Sign In</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <link rel="stylesheet" href="{% static '/style/signIn.css' %}">
</head>
<body>
    <form method="post">
        <h2>Sign In</h2>
        {% csrf_token %}
        <label for="username">Username:</label>
        <input type="text" name="username" required>
        <label for="password">Password:</label>
        <input type="password" name="password" required>
        <button type="submit">Sign In</button>
        {% if error %}
            <p>{{ error }}</p>
        {% endif %}
    </form>
</body>
</html>
```

signup.html

```
<!DOCTYPE html>
<html>
<head>
    {% load static %}
    <title>Sign Up</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <link rel="stylesheet" href="{% static '/style/signUp.css' %}">
</head>
<body>
```


```

    <h2>Sign up</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Sign up</button>
    </form>
</body>
</html>

```

shopCDB/man.html

```

{% load static %}
{% include '../style.html' %}
{% include 'nav.html' %}
{% block man %}
<main>
<h1>Man</h1>
{% block manshow %}
<div class="products-container">
    {% for product in products %}
        <div class="product rounded-corners">
            
            <h3>{{ product.name }}</h3>
            <p>Giá: {{ product.price }}$</p>
            <button class="favorite-button">{{ product.favorite|yesno:"Favorited,Add to
favorites" }}</button>
            <a href="{% url 'product_detail' product.id %}" class="more-info-
button">Show</a>
            <button class="buy-now-button">Buy now 

```

shopCDB/women.html

```

{% load static %}
{% include '../style.html' %}
{% include 'nav.html' %}
{% block women %}
<main>
<h1>Women</h1>

```

```

        {% block search %}
        <div class="products-container">
            {% for product in shop_women %}
            <div class="product rounded-corners">
                <!-- Đảm bảo rằng 'product.image' có giá trị và 'product.image.url' trả về đường
dẫn đúng -->
                

                <h3>{{ product.name }}</h3>
                <p>Giá: {{ product.price }}$</p>
                <a href="{% url 'product_detail' product.id %}" class="more-info-button">Show</a>

                <form action="{% url 'add_to_cart' %}" method="post">
                    {% csrf_token %}
                    <input type="hidden" name="product_id" value="{{ product.id }}">
                    <input type="submit" value="Buy now 🛒" class="buy-now-button">
                </form>
            </div>
        {% empty %}
            <p>No products available.</p>
        {% endfor %}
        </div>
        {% endblock search %}
    </main>
    {% endblock women %}
    {% include '../footer.html' %}

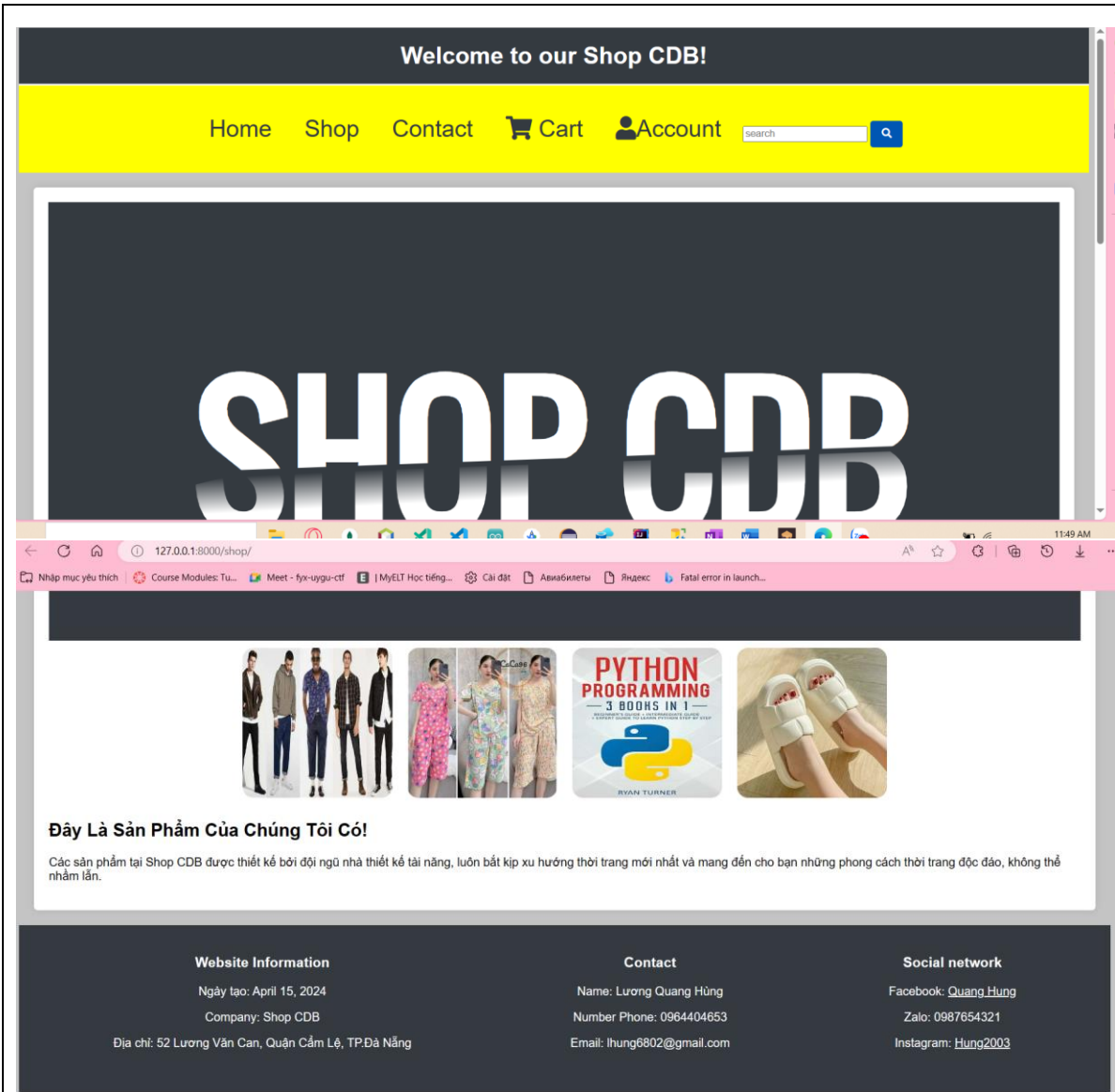
```

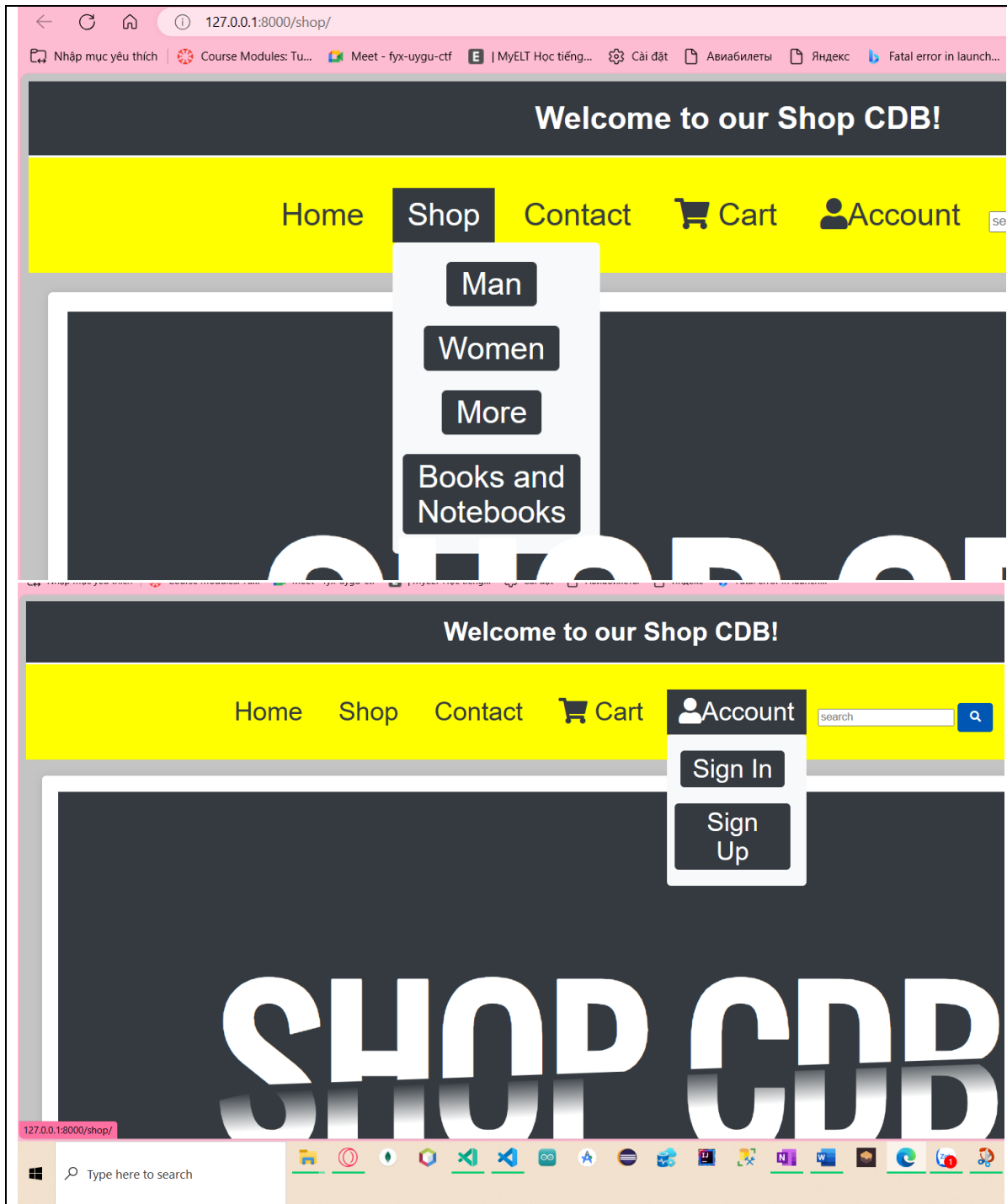
shopCDB/products.html

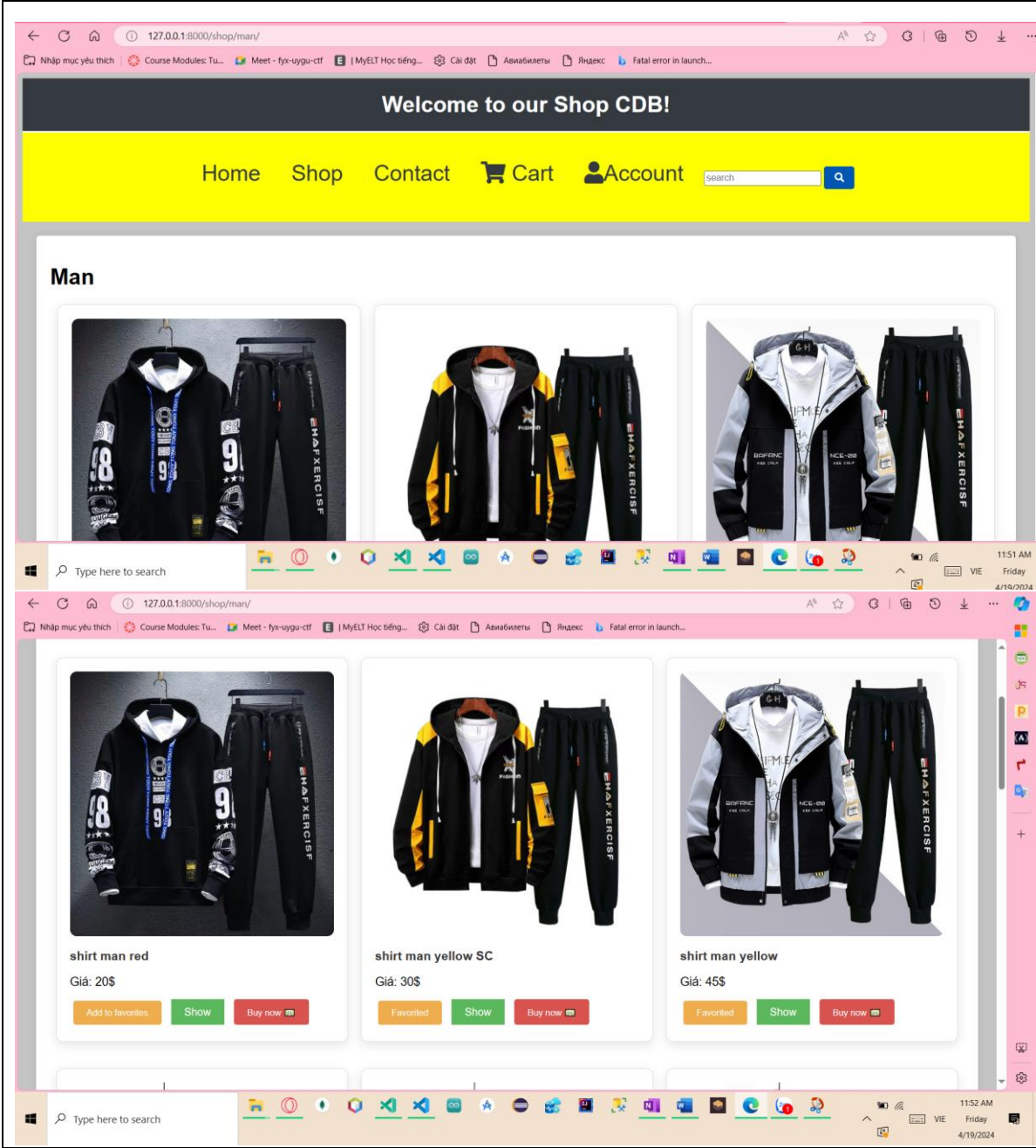
```

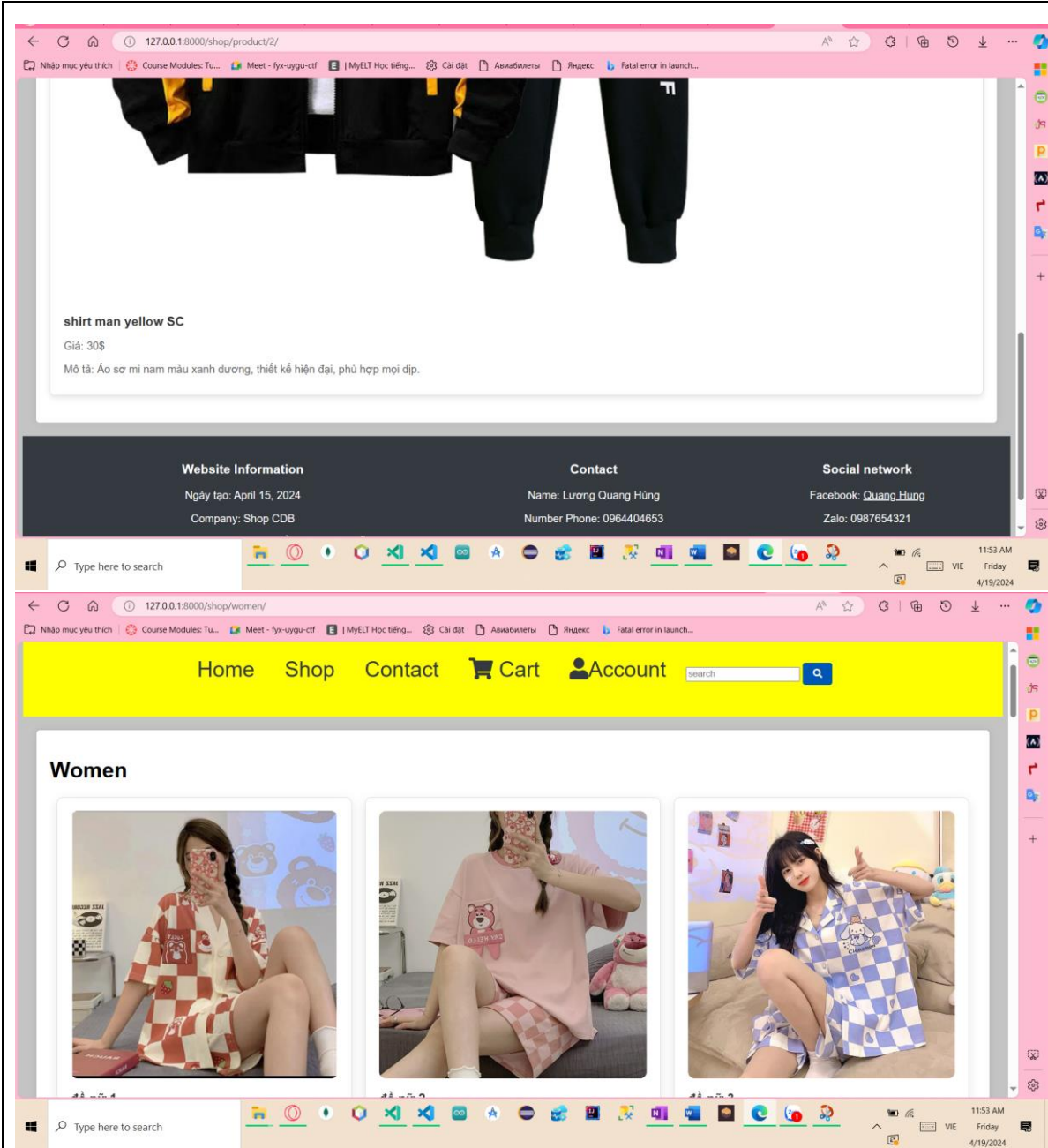
{% load static %}
{% include '../style.html' %}
{% include 'nav.html' %}
<main>
{% block manshow %}
<div class="product-detail">
    
    <h3>{{ product.name }}</h3>
    <p>Giá: {{ product.price }}$</p>
    <p>Mô tả: {{ product.description }}</p>
    <!-- Thêm bất kỳ thông tin chi tiết nào khác bạn muốn hiển thị ở đây -->
</div>
{% endblock %}
</main>
{% include '../footer.html' %}

```












đồ nữ 3

HomeShopContactCartAccount

Kết quả tìm kiếm




đồ nữ 3

Giá: 50.00\$

127.0.0.1:8000/shop/search/?q=đồ+nữ+3

HomeShopContactCartAccountcon bướm sinh

Kết quả tìm kiếm



Welcome to our Shop CDB!

HomeShopContactCartAccountcon bướm sinh

Kết quả tìm kiếm

Không tìm thấy sản phẩm nào.

Website Information

Ngày tạo:

Company:

Địa chỉ:

Contact

Name:

Number Phone:

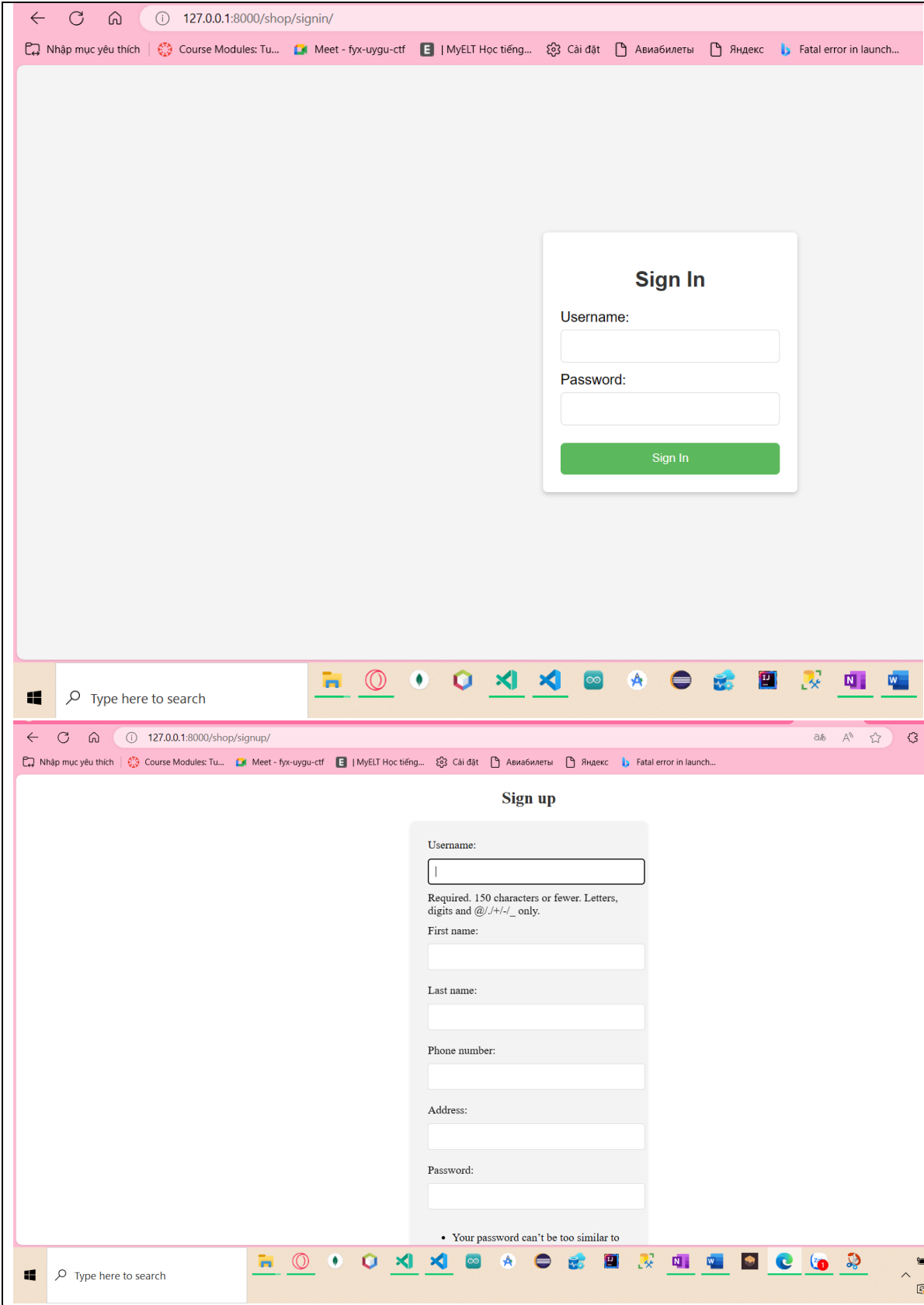
Email:

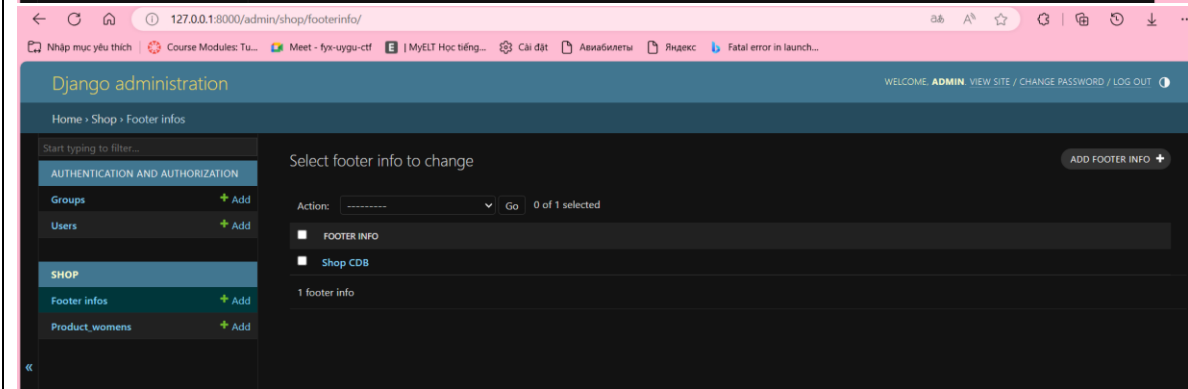
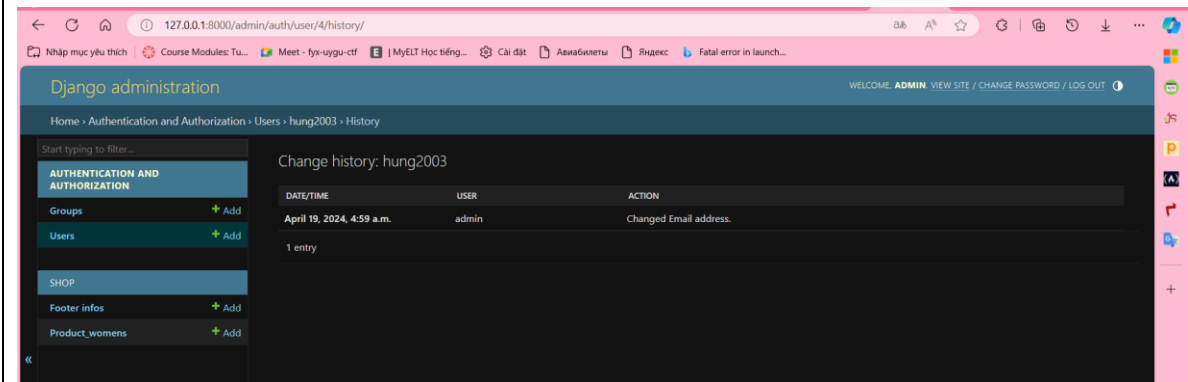
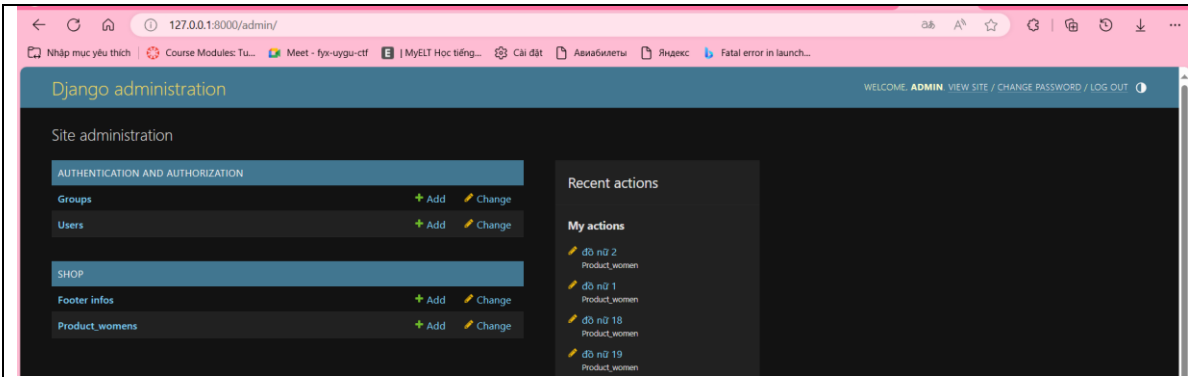
Social network

Facebook: [Quang Hung](#)

Zalo:

Instagram: [Hung2003](#)





127.0.0.1:8000/admin/shop/footerinfo/1/change/

Nhập mục yêu thích

Course Modules: Tu...

Meet - fyx-uygu-ctf

E | MyELT Học tiếng...

Cài đặt

Авиабилеты

Яндекс

Fatal error in launch...

Home > Shop > Footer infos > Shop CDB

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

SHOP

Footer infos + Add

Product_womens + Add

Change footer info

Shop CDB

Ngày tạo: 2024-04-15 Today | 📅

Note: You are 7 hours ahead of server time.

Tên công ty: Shop CDB

Địa chỉ: 52 Lương Văn Can, Quận Cẩm Lệ, TP.Đà Nẵng

Họ tên: Lương Quang Hùng

Số điện thoại: 0964404653

Email: lhung6802@gmail.com

Facebook:

Currently: https://facebook.com/

Change: https://facebook.com/

Zalo: 0987654321

Instagram:

Currently: https://instagram.com/

Change: https://instagram.com/

Django administration

WELCOME: ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT ⓘ

Home > Shop > Product_womens > đồ nữ 5

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

SHOP

Footer infos + Add

Product_womens + Add

Change product_women

đồ nữ 5 HISTORY

Name: đồ nữ 5

Image:

Currently: products/do-nu-05.jpg

Change:

Chọn tệp

 không tệp nào được chọn

Description:

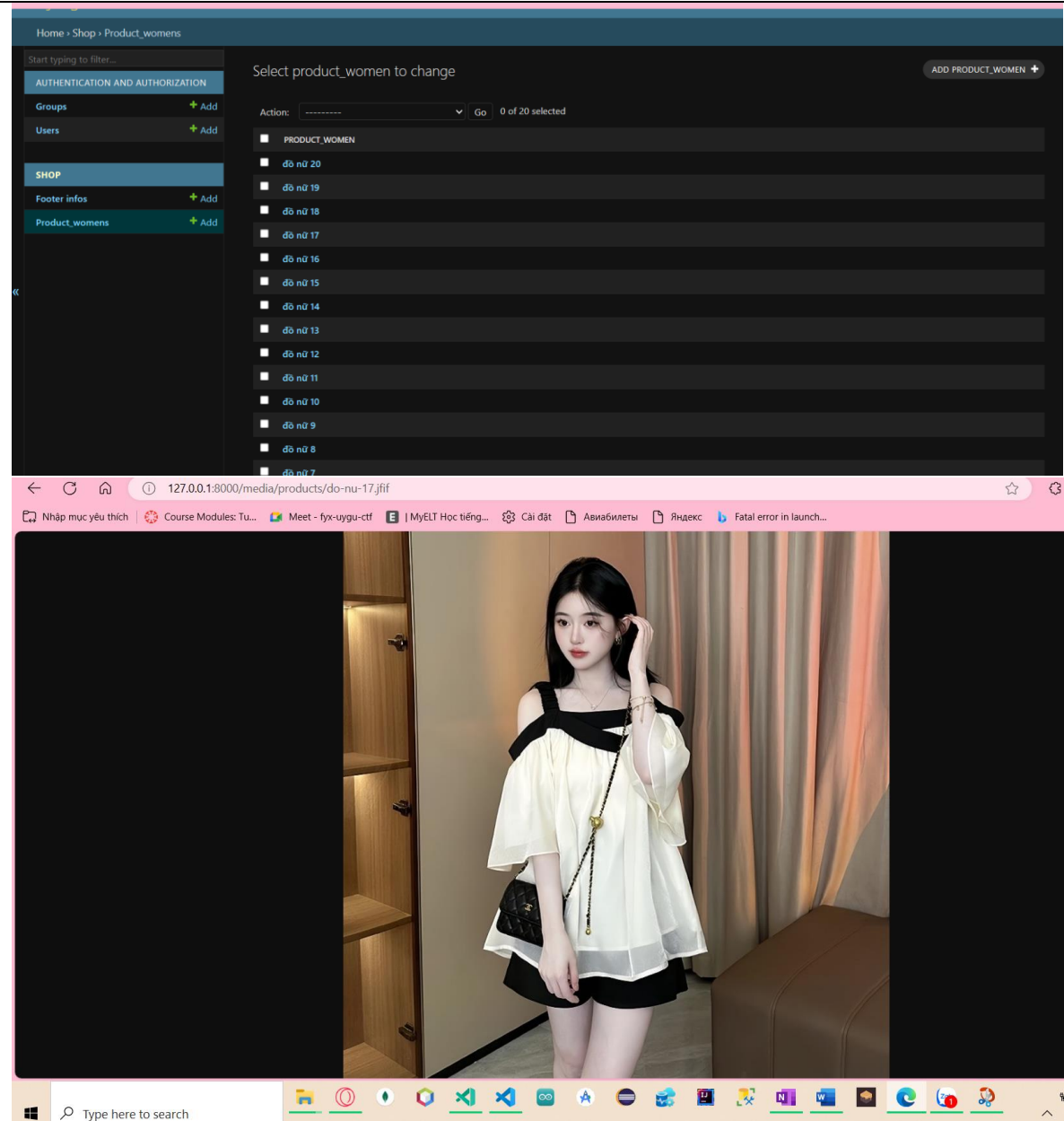
LƯU Ý : CHỈ ĐẶT CHUẨN SIZE ĐỂ VỪA CHÂN VÁY, ÁO FORM RẤT RỘNG, K ĐẶT CHỈNH SIZE

Chân váy có độ 4 size : S M L XL (khách lấy đúng theo số đo eo)

Áo form to nên chỉ có 2 size : số 1 (tương đương S M) và số 2 (tương đương L XL)

Price: 50,00

Quantity: 20



End
Thank you