KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ 1, NĂM HỌC 2023-2024**

**ĐỀ TÀI:**

**SỬ DỤNG DJANGO ĐỂ PHÁT TRIỂN ỨNG DỤNG WEBSITE BÁN QUẦN ÁO THỂ THAO**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn**  ĐOÀN PHƯỚC MIỀN | **Sinh viên thực hiện**  TRẦN PHÚC VĨ  Mã Lớp DA20TTA  MSSV 110120084  Khóa 2020-2024 |
|  |  |

**Trà Vinh, tháng 2 năm 2023**

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ 1, NĂM HỌC 2023-2024**

**ĐỀ TÀI:**

**SỬ DỤNG DJANGO ĐỂ PHÁT TRIỂN ỨNG DỤNG WEBSITE BÁN QUẦN ÁO THỂ THAO**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn**  ĐOÀN PHƯỚC MIỀN | **Sinh viên thực hiện**  TRẦN PHÚC VĨ  Mã Lớp DA20TTA  MSSV 110120084  Khóa 2020-2024 |
|  |  |

**Trà Vinh, tháng 2 năm 2023**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**Đoàn Phước Miền**

**LỜI CẢM ƠN**

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

Tôi xin chân thành cảm ơn quí thầy cô đã giúp đỡ tôi thực hiện đề tài này. Đặc biệt thầy **Đoàn Phước Miền** đã tận tình hướng dẫn, giúp đỡ, chỉ bảo tôi trong suốt thời gian thực hiện thực tập đồ án chuyên ngành.

Đồng thời tôi cũng xin trân trọng cảm ơn những tình cảm quí báu mà các thầy cô trong trường Đại Học Trà Vinh đã truyền đạt cho tôi, những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đề tài này.

Tuy nhiên, do khả năng đọc hiểu cũng như vận dụng kiến thức vào bài làm chưa được chắc chắn nên tôi không thể phát huy hết những ý tưởng, khả năng hỗ trợ của ngôn ngữ và kỹ thuật lập trình vào đề tài. Trong quá trình xây dựng website không thể tránh khỏi những sai sót, mong quí thầy cô đóng góp và cảm thông.

Tôi xin chân thành cảm ơn.

***Trà Vinh,*** *ngày….tháng….năm….*

**Sinh viên thực hiện**

**Trần Phúc Vĩ**

**MỤC LỤC**

**DANH MỤC TỪ VIẾT TẮT**

**DANH MỤC HÌNH ẢNH**

**TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH**

**MỞ ĐẦU**

Trong thời đại công nghệ hiện nay, việc mua bán trên Internet đã trở nên phỗ biến, trở thành một phần không thể thiếu của đời sống, việc nắm bắt tìm hiểu, phân tích nhu cầu tiêu dùng là một phần quan trọng, nó giúp người tiêu dùng dễ dàng chọn lựa sản phẩm phù hợp, giúp các nhà kinh doanh nắm bắt được thị hiếu các sản phẩm có trên thị trường mà khách hàng mong muốn và đáp ứng.

Khi xã hội ngày càng phát triển, mức sống của người dân được nâng cao thu nhập kinh tế ngày càng được cải thiện thì việc mua bán các sản phẩm trên Internet đã trở thành một phần không thể thiếu trong đời sống. Người tiêu dùng có thể dễ dàng đặt mua các mặt hàng mà mình yêu thích chỉ với vài thao tác trên website, giúp tối ưu thời gian hơn khi phải đến tận cửa hàng chọn lựa.

Ngoài ra, phát triển trang web có các tính năng nâng cao giúp tối ưu hóa trải nghiệm website của người dùng cũng là cách giúp website ngày càng phát triển một cách bền vững và hiệu quả.

Do đó tôi chọn thực hiện đề tài “Sử dụng DJANGO để phát triển ứng dụng website bán quần áo thể thao” với mục đích cung cấp một website tiện ích giúp cho người dùng có thể mua bán hàng hóa một cách thuận tiện nhất.

Để xây dựng website bán hàng bằng DJANGO, trước hết tôi tìm hiểu về DJANGO, cài đặt ứng dụng Visual Studio Code, và ngôn ngữ Python để tiến hành chạy code và thêm DJANGO vào bài code của mình.

# CHƯƠNG 1: TỔNG QUAN

## 1.1 Mô tả bài toán

Hiện nay, sự phát triển của công nghệ tiên tiến ngày càng mạnh mẽ và ứng dụng của nó ngày càng rộng rãi hơn. Trước đó khi chưa đạt đến trình độ công nghệ như bây giờ, người dân thật sự phải tốn rất nhiều thời gian cũng như công sức để có thể đi đến các nơi mua sắm. Tầm quan trọng của công nghệ hiện nay là không thể thay thế được.

Trong cuộc sống ngày càng bận rộn như hiện nay, để giảm bớt các bất cập trong việc phải đến các cửa hàng, các shop quần áo thể thao để mua quần áo mà mình yêu thích, các website bán quần áo thể thao ngày càng nhiều và trở nên phổ biến, nắm bắt cơ hội này việc xây dựng một website bán quần áo thể thao trực tuyến sẽ giúp khách hàng giảm bớt thời gian và công sức đến cửa hàng mua hàng.

Để thiết kết website bán quần áo thể thao ta cần có những công nghệ cũng như các phần mềm để viết ra trang web. Tôi sử dụng Visual Studio Code với Django để thiết kế ra một website bán quần áo thể thao. Vì Django là một framework mạnh nên tôi kết hợp giữa HTML, CSS làm giao diện và Django xử lý các chức năng như: đăng ký, đăng nhập, thêm thông tin sản phẩm vào giao diện, thêm sản phẩm vào giỏ hàng,sửa , xóa sản phẩm,…

## 1.2 Mục tiêu hướng, giải quyết và kế hoạch thực hiện

### 1.2.1 Mục tiêu cần đạt được

Am hiểu Django framework căng bản.

Hoàn thiện trang website giúp khách hàng có thể tự do mua các mặc hàng mà mình yêu thích và phù hợp với mọi lứa tuổi, với những chức năng cơ bản như:

1. Hiển thị các sản phẩm theo từng danh mục.
2. Đăng nhập vào hệ thống: nhập thông tin tài khoản, mật khẩu để vào hệ thống. Phân quyền theo nhu cầu của người dùng.
3. Quản lý hàng hóa: thêm, xóa, sửa thông tin các loại hàng hóa và các mặt hàng, tìm kiếm các thông tin hàng hóa.
4. Cung cấp cho khách hàng những cập nhật mới nhất về sản phẩm và giá cả.
5. Giao diện tiện lợi, dễ dàng sử dụng.
6. Quản lý giỏ hàng: xóa, sửa số lượng các loại hàng hóa

### 1.2.2 Hướng giải quyết và kế hoạch thực hiện

**Hướng giải quyết**

- Tham khảo các diễn đàn các website về bán quần áo thể thao khác.

- Liên tục cập nhật dữ liệu mới, mặt hàng mới.

- Học hỏi và đổi mới website bằng cách tham khảo những website bán điện thoại hiện có trên thị trường.

- Thiết kế các giao diện của website.

- Cài đặt chương trình ứng dụng, nhập liệu, chạy thử và kiểm tra lỗi.

- Viết một bài báo cáo về công việc đã thực hiện theo mẫu quy định

**Kế hoạch thực hiện**

- Tìm hiểu về Django, các chức năng, lợi ích của việc sử dụng Django trong việc xây dựng một website bán hàng.

- Khảo sát yêu cầu của khách hàng.

- Tiến hành tìm hiểu, nghiên cứu cách hoạt động của một website bán hàng.

- Tham khảo một số trang web bán điện thoại hiện có trên thị trường như: Coolmate, Sporter.vn, …

- Tiến hành các bước cơ bản, thiết kế giao diện.

- Làm các chức năng cho trang web.

- Chạy thử và kiểm lỗi.

- Hoàn thiện trang web bán quần áo thể thao.

**Môi trường cài đặt:**

Để thực hiện được yêu cầu phải cần có một ứng dụng lập trình đáp ứng được các chức năng thiết kế, cùng với các thư viện cơ bản hỗ trợ.

- Ứng dụng: Visual Studio Code.

- Sử dụng HTML, CSS, JavaScript, thư viện: Django framework.

# **CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT**

## Khái niệm Python

Python là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và dễ học. Guido van Rossum phát triển Python vào những năm 1990, và từ đó, ngôn ngữ này đã trở thành một trong những ngôn ngữ phổ biến nhất trên toàn cầu.

Tính linh hoạt và đa dụng là những điểm mạnh của Python. Nó có khả năng thực hiện nhiều loại công việc khác nhau, từ viết các script đơn giản cho đến xây dựng các ứng dụng web phức tạp.

Cú pháp của Python rất dễ đọc và gần gũi với ngôn ngữ tự nhiên. Điều này làm cho việc viết và đọc mã trở nên dễ dàng hơn, giúp tăng tốc quá trình phát triển và hiểu mã nguồn của người khác.

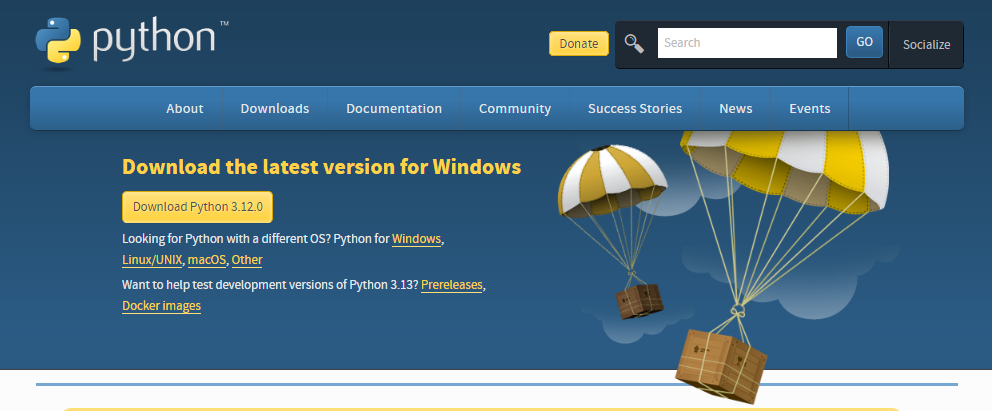
Cộng đồng Python rất lớn, đa dạng và sôi động. Cộng đồng này không chỉ đóng góp các thư viện và framework hữu ích mà còn cung cấp sự hỗ trợ cho nhau, từ người mới học đến những chuyên gia hàng đầu.

Một số điểm nổi bật của Python:

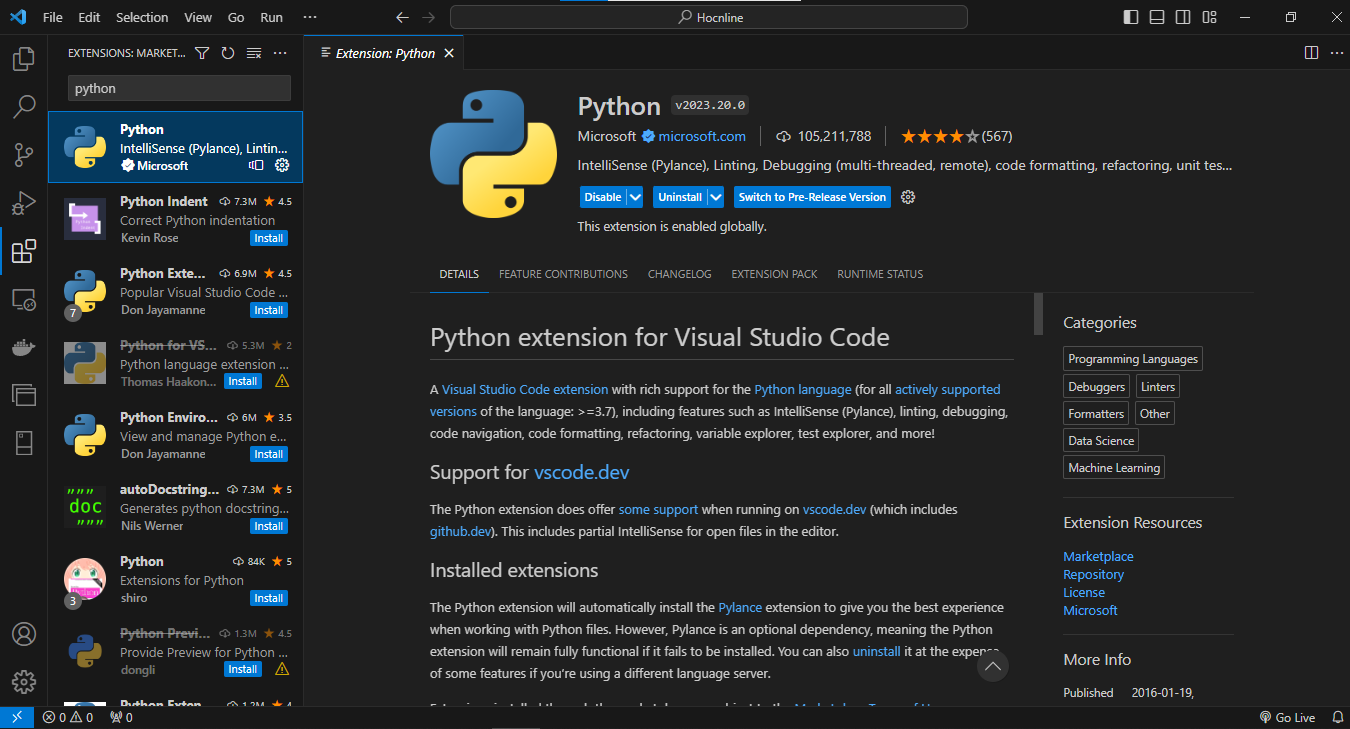
* Python có cú pháp rõ ràng và gần gũi với ngôn ngữ tự nhiên, giúp người dùng dễ dàng tiếp cận và hiểu code.
* Nó được sử dụng cho nhiều loại dự án, từ phát triển web, xử lý dữ liệu, machine learning cho đến hacking và tự động hóa công việc.
* Python có một cộng đồng lớn và đa dạng, cung cấp tài liệu phong phú, thư viện và sự hỗ trợ từ cộng đồng người dùng.
* Nó cung cấp một loạt các công cụ và frameworks mạnh mẽ giúp giải quyết nhiều vấn đề khác nhau.
* Python có khả năng di động và có thể chạy trên nhiều hệ điều hành và nền tảng khác nhau.

### Cách cài đặt Python

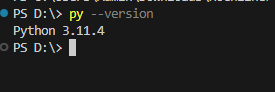
Đầu tiên, truy cập vào địa chỉ <https://www.python.org/downloads/> và tải bản Python về máy



Sau đó ta tiếp tục cài đặt Python vào Visual Studio Code để tiến hành viết dự án Django, khởi động Visual Studio Code vào phần **Extensions** sau đó gõ vào Python và ấn **install**



Để kiểm tra xem Python đã được cày đặt thành công hay chưa, ta vào terminal của Visual Studio Code và gõ py –version



## 2.2 Khái niệm về Django

Django là framework giúp cho người dùng có thể sử dụng để phát triển các ứng dụng web một cách nhanh chóng và hiệu quả. Hầu hết các ứng dụng web có một số chức năng phổ biến, như xác thực, truy xuất dữ liệu từ cơ sở dữ liệu và quản lý cookie. Người dùng phải viết mã cho chức năng tương tự vào mọi ứng dụng web mà họ viết. Django giúp họ làm việc dễ dàng và nhanh chóng hơn bằng cách nhóm các chức năng khác nhau thành một tập hợp lớn các mô-đun có thể tái sử dụng, được gọi là một khung ứng dụng web. Người dùng sử dụng khung web Django để sắp xếp và viết mã của họ hiệu quả hơn và giảm đáng kể thời gian phát triển web.

Django tập trung vào hai khía cạnh quan trọng là tính "tái sử dụng" và "tự chạy", đồng thời cung cấp khả năng phát triển nhanh và tránh việc làm lại công việc đã có sẵn. Framework này được thiết kế với mục tiêu chuyển ý tưởng thành sản phẩm một cách nhanh nhất có thể. Bằng cách kết hợp hoàn hảo các tính năng này, Django cho phép chúng ta xây dựng các ứng dụng như website bán hàng hoặc hệ thống quản lý hàng hóa với độ chi tiết và độ chính xác cao một cách hiệu quả..

### 2.2.1 Tính chất của Django

Độ hoàn thành cao:

* Django cung cấp hầu hết mọi thứ mà các developer có thể muốn để phát triển web theo hướng mình muốn.
* Tất cả các phần trong framework hoạt động liền mạch với nhau, tuân theo một nguyên tắc thiết kế nhất quán và có tài liệu để bạn tham khảo. Nhờ đó, bạn có thể vừa phát triển web theo phong cách riêng, vừa tiết kiệm được thời gian

Linh hoạt:

* Django có thể sử dụng để xây dựng hầu hết mọi loại trang web- từ hệ thống quản lý nội dung (như wiki), cho đến các trang mạng xã hội, tin tức.
* Nó có thể hoạt động cùng với các framework bên ngoài và cũng có thể cung cấp nội dung ở hầu hết mọi định dạng (bao gồm HTML, RSS feeds, JSON, XML,..v..v..).
* Framework này còn cung cấp các tùy chọn khác nhau cho hầu hết các chức năng (như công cụ tạo template, cơ sở dữ liệu phổ biến,..).

Bảo mật:

* Django giúp developer tránh được nhiều lỗi bảo mật phổ biến bằng cách cung cấp framework có khả năng tự bảo vệ trang.
* Django còn bảo vệ website khỏi những lỗ hổng khỏi những loại tấn công mạng như: tấn công SQL injection, Cross-site Scripting, cross-site request forgery và clickjacking.

Khả năng mở rộng:

* Django sử dụng kiến trúc thành phần riêng nên bạn có thể mở rộng quy mô bằng cách thêm phần cứng vào các cấp độ (máy chủ bộ nhớ đệm, máy chủ cơ sở dữ liệu hoặc máy chủ ứng dụng).

Khả năng duy trì:

* Code của Django được viết bằng cách sử dụng các nguyên tắc và mẫu thiết kế khuyến khích việc tạo mã có thể bảo trì và tái sử dụng.
* Nó còn thúc đẩy việc nhóm các chức năng liên quan thành các “ứng dụng” có thể tái sử dụng, từ đó giúp website có khả năng duy trì cao hơn.

### 2.2.2 Các tính năng cơ bản

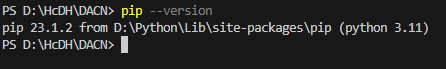
Dưới đây là một số tính năng cơ bản và thường sử dụng:

* **Admin Interface tự động**: Django cung cấp một giao diện quản trị tự động để quản lý dữ liệu của ứng dụng mà không cần phải viết code. Bằng cách định nghĩa các models, Django tạo ra một giao diện quản trị sẵn có để thêm, sửa, xóa dữ liệu.
* **URL Routing**: Django sử dụng **URLconf (URL configuration)** để ánh xạ các URL của ứng dụng đến các views tương ứng.
* **Templates**: Django cung cấp hệ thống template để xây dựng giao diện người dùng.
* **Authentication và Authorization**: Django cung cấp các công cụ mạnh mẽ để xác thực người dùng và kiểm soát quyền truy cập thông qua hệ thống user và groups.
* **ORM (Object-Relational Mapping):** Django cung cấp một ORM mạnh mẽ cho phép tương tác với cơ sở dữ liệu thông qua việc sử dụng Python thay vì việc viết các truy vấn SQL trực tiếp.

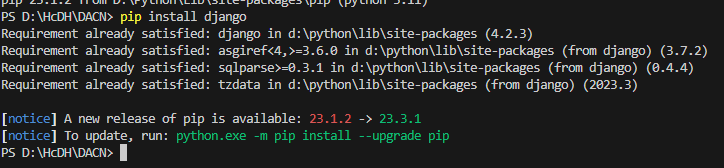
### 2.2.3 Cài đặt Django

Django là Framework mạnh mẽ, giúp việc xây dựng website dễ dàng và tiện lợi hơn, để cài đặt Django, trước hết ta cần cài đặt pip để có thể chạy câu lệnh cài đặt Django. Để cài đặt pip ta vào <https://pip.pypa.io/> làm theo hướng dẫn và cài đặt pip

Để kiểm tra xem pip đã cài đặt thành công chưa, ta chạy lại pip –version



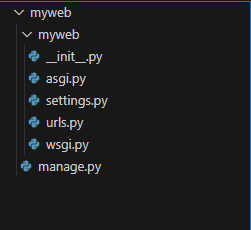
Sau khi đã cài đặt pip thành công ta tiến hành cài đặt Django bằng câu lệnh “**pip install Django**”



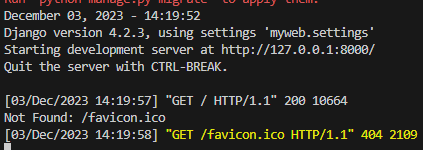
### 2.2.4 Tạo project đầu tiên

Để tạo Project với Django trước tiên ta vào thư mục cần để lưu dự án, sau đó bật cửa sổ terminal của Visual Studio Code lên và nhập vào câu lệnh “**django-admin startproject nameproject**”, nameproject là tên dự án muốn tạo.

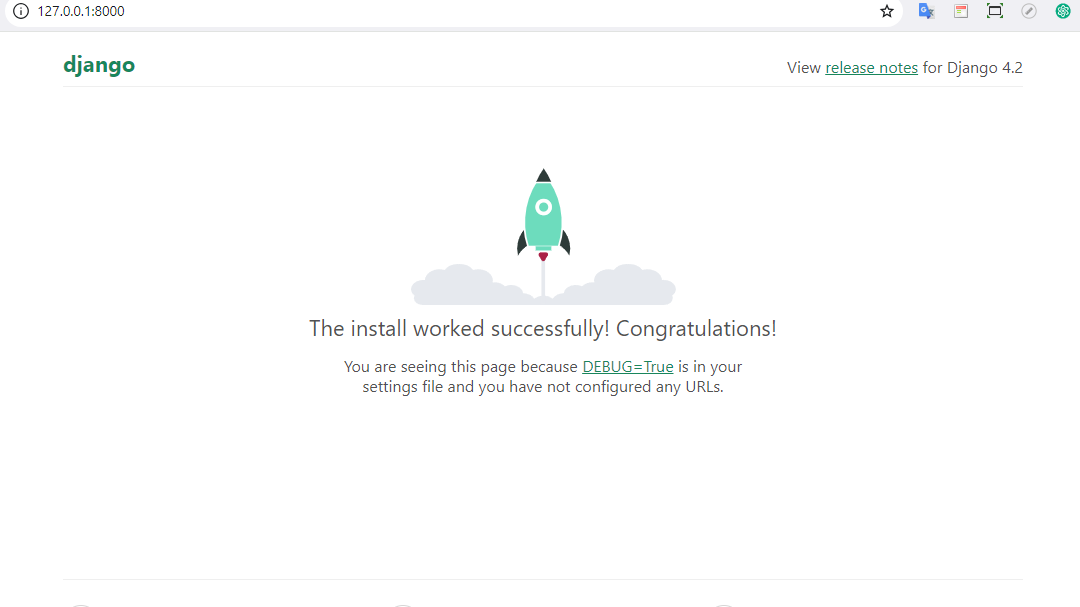
Sau khi tạo thành công, dự án với tên “**myweb**” có cấu trúc như sau:



Để khởi chạy dự án vừa tạo, ta sử dụng câu lệnh “**python manage.py runserver**”



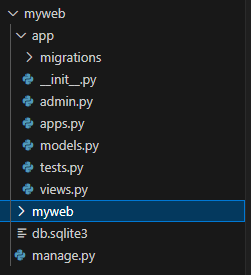
Sau đó ta truy cập vào đường dẫn <http://127.0.0.1.8000/> để xem kết quả đã thành công chưa.



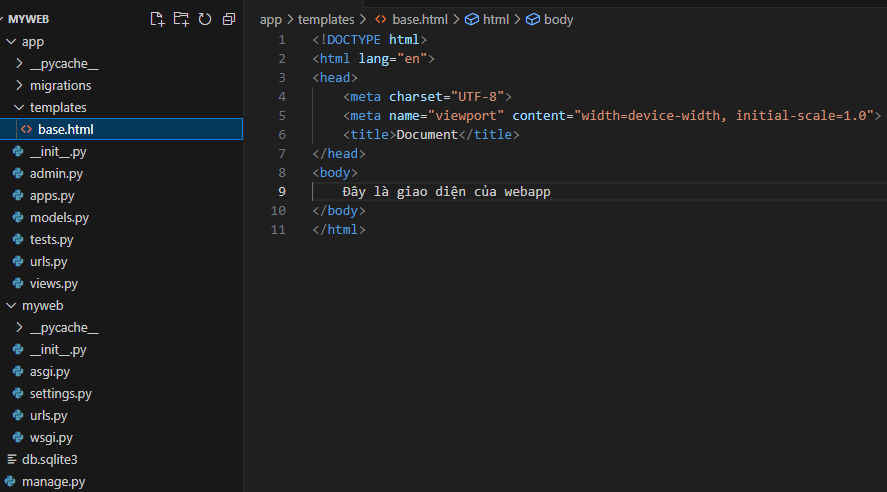
Để có thể tạo một dự án hoàn chỉnh, ta cần tạo các web app để khởi chạy dự án, web app là một project bao gồm nhiều thư mục để khởi chạy dự án, trong đó mỗi web app đảm nhận một công việc riêng và chứa các file chuẩn của một ứng dụng web Django.

Để tạo web app, ta sử dụng câu lệnh “python manage.py startapp nameapp”, nameapp là tên app cần được tạo.

Sau khi tạo thành công, đây là cấu trúc của một web app

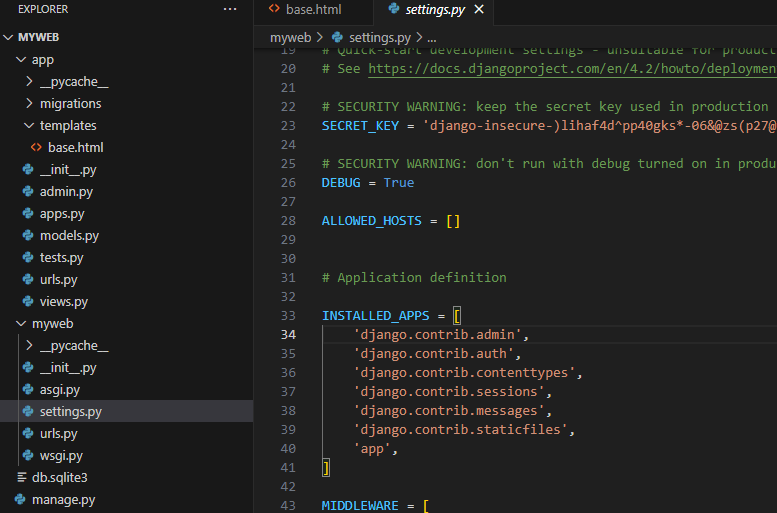


Để có thuận tiện cho việc quản lí dự án, trong web app, ta tạo một thư mục có tên là **templates** dùng để lưu giao diện của hệ thống web, bên trong thư mục **templates** là các file **HTML** dùng để xuất ra giao diện của web

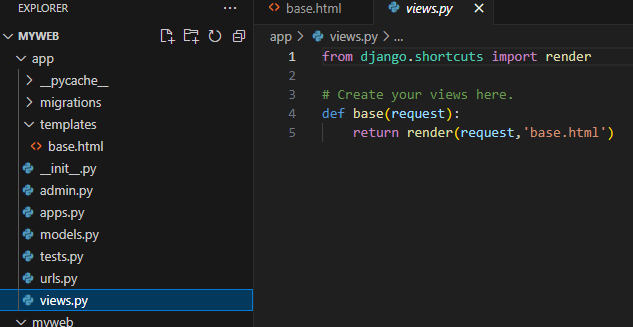
****

Ngoài ra để web có thể hoạt động được, ta cần khai báo web app vừa tạo trong file **settings.py** của thư mục gốc của dự án, ở đây là thư mục **myweb**

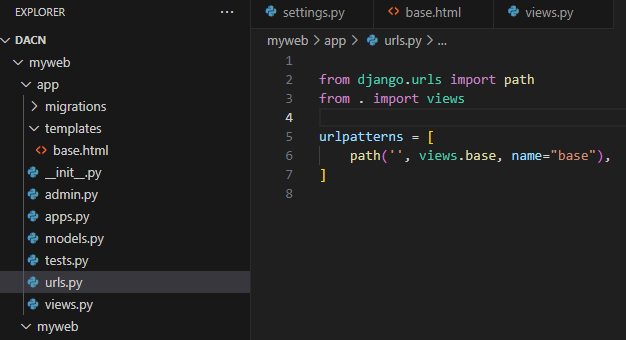
Vào **settings.py** kéo xuống đến phần  **INSTALLED\_APPS** và khai báo web app ở đây.

****

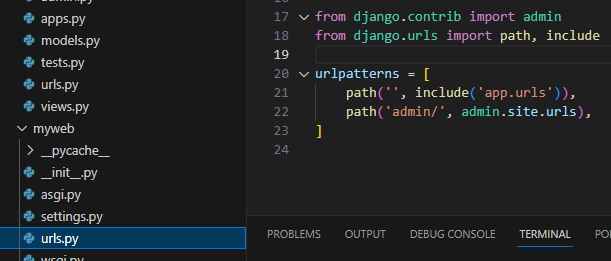
Tiếp theo, vào file **views.py** của web app, ở đây ta cần khai báo các **templates** cần được sử dụng.



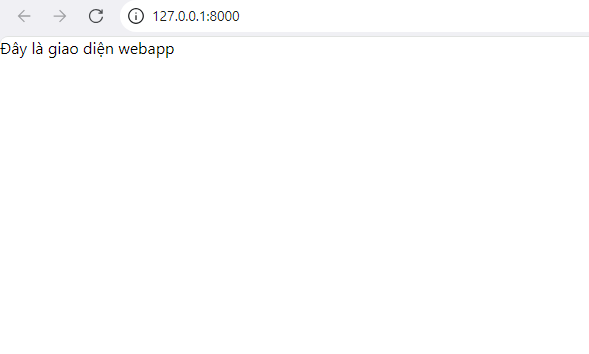
Sau đó web cần file có tên là **urls.py** để có thể hoạt động, ở đây là nơi lưu địa chỉ dẫn đến các **templates** đã được khai báo trong file **views.py**.



Ngoài ra trong file **urls.py** của thư mục gốc, ta cần khai báo file **urls.py** của webapp để có thể vận hành web.



Để khởi chạy web, ta vào terminal của Visual Studio Code, nhập vào câu lệnh **python manage.py runserver**

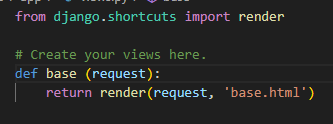


### 2.2.5 Các thành phần chính có trong Django

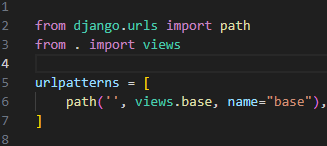
Trong phần trên, các thành phần của Django chỉ được nói sơ lượt qua chứ không cụ thể từng phần, ở trong phần này, ta sẽ tìm hiểu kĩ hơn về các thành phần chính của Django.

**1. Views**

Là các hàm, hay các lớp xử lý logic của ứng dụng, nhận yêu cầu URL từ người dùng, khi người dùng vào các trang khác nhau, **views** sẽ trỏ đến các hàm tương ứng để phản hồi lại người dùng.Trong Django khi một trang web được tạo ra trong file **views.py** nó sẽ được gọi đến tuỳ thuộc vào URL mà ta đã chọn khi khai báo trong file **urls.py** , khái niệm này được gọi là URLConf, đây là một module Python của Django làm nhiệm vụ phân tích và dẫn truyền đến một hàm View nhất định, đây là một hàm cơ bản trong **Views**



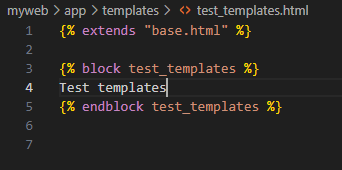
Cấu trúc của file **urls.py** gồm đoạn đầu để **import** các thư viện cần thiết, tiếp theo là phần **urlpatterns** là nơi chứa có liên kết dẫn đến các trang trong web. Cấu trúc của khai báo đường dẫn cơ bản gồm: path(‘đây là tên của địa chỉ khi hiển thị trên web’, đây là nơi khai báo các **templates** đã được khai báo ở file **views.py** , name=’đây là tên của liên kết’).

****

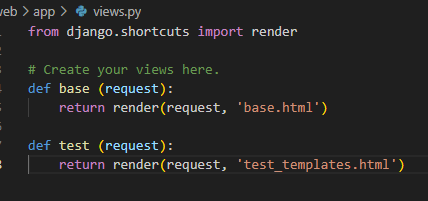
**2. Templates**

Templates trong Django là các layout web sẵn có, giúp tiết kiệm thời gian phát triển trang web bằng cách chỉ cần thêm nội dung chính vào chúng. Trình duyệt chỉ hiểu và hiển thị code HTML, không hiểu code Python. Để sử dụng code Python trong trang web, Django cung cấp các thẻ template, được đánh dấu bắt đầu và kết thúc bằng cặp kí tự {% %} hoặc {{ }}. Các câu lệnh Python nằm trong cặp {% %}, trong khi các biến được đặt trong cặp {{ }}. Ngôn ngữ Template của Django được thiết kế để hỗ trợ những người đã có kinh nghiệm làm việc với HTML, vì vậy nếu bạn đã quen với HTML, việc làm quen với Template trong Django sẽ không quá khó khăn.

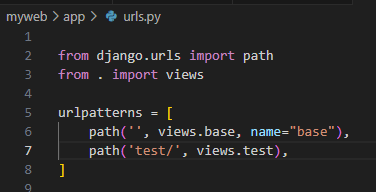
Ở đây trang **base.html** là phần giao diện tĩnh, vào trang **test\_templates.html** là phần nội dung được hiển thị khi ta gọi đến.



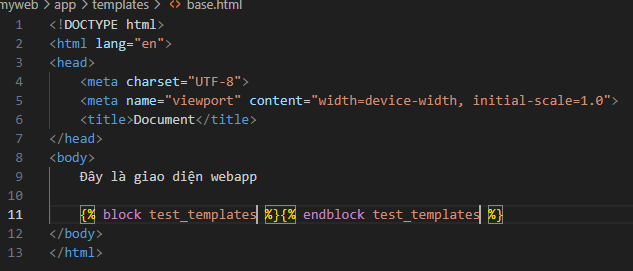
Tại file **views.py** ta tạo một hàm để gọi đến trang, ở đây hàm có tên là **test**



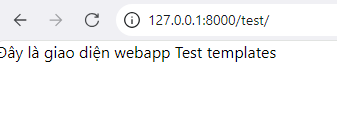
Ta vào **urls.py** để cấu hình đường dẫn đến hàm test trong View



Để có thể hiển thị nội dung của trang **test\_templates.html** lên trang **base.html** ta sử dụng cặp dấu ngoặc {%%} như sau:



Đây là kết quả, như ta thấy khi trỏ đến đường dẫn **test/** giao diện của web sẽ có thêm nội dung của trang **test\_templates.html**

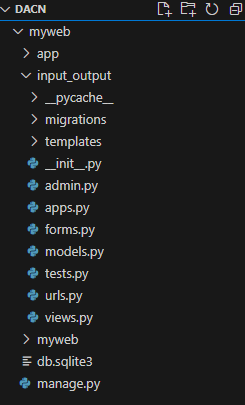


**3. Models**

Ở phần **Models** này sẽ có một ví dụ khác so với các ví dụ đơn giản ở trên, nhưng trước hết hãy tìm hiểu **Models** trong **Django** là gì:

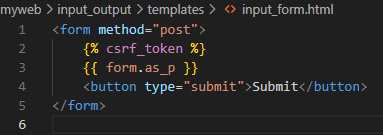
Trong **Django**, **Models** được dùng để định nghĩa và cấu trúc nên cơ sở dữ liệu bằng cách sử dụng các lớp trong Python. Mới lớp tương ứng là một bảng trong cơ sở dữ liệu, còn các trường bên trong tương ứng với các cột trong cơ sở dữ liệu. Giúp cho cơ sở dữ liệu có thể linh hoạt hơn khi chỉ cần chạy lệnh sẽ tự tạo ra cơ sỡ dữ liệu.

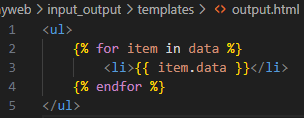
Trong ví dụ này, ta sẽ tạo mới một webapp có tên là **input\_output** và tạo ra thư mục **templates** chứa giao diện của webapp.



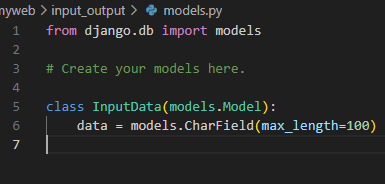
Trong thư mục **templates** tạo ra 2 file giao diện với tên gọi là **input.html** và **output.html**

Đây là phần code của 2 giao diện để có thể nắm rõ các tương tác cũng như các dữ liệu nhập vào đâu và xuất ra ở đâu.

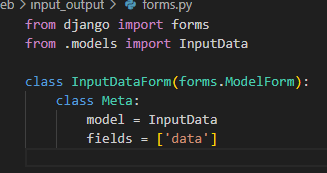




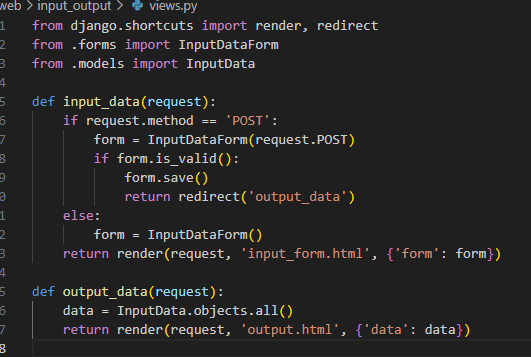
Sau đó ta vào **Models** để định nghĩa một model dùng để lưu trữ dữ liệu. Tạo ra một trường **data** với kiểu dữ liệu Char và độ dài 100



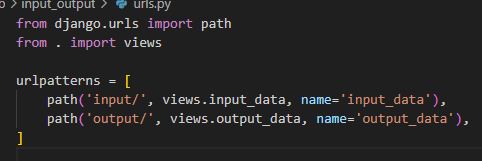
Tạo một form để nhập dữ liệu trong **form.py**



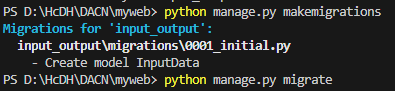
Tiếp theo ta cấu hình trong **Views**. Như chúng ta thấy, hàm **input\_data** sẽ gọi đến phương thức **POST** để xác nhận dữ liệu nhập vào và lưu vào **form,**  xét điều kiện nếu tồn tại dữ liệu sẽ gọi ra biến save() để lưu vào cơ sở dữ liệu và return về **output\_data.** Ngược lại nếu request không phải là phương thức **POST** thì sẽ gọi ra **InputDataForm()** để có thể nhập giá trị vào. Còn ở hàm **output\_data** sẽ lấy tất cả cả dữ liệu của hàm **InputData** và gọi đến trang **output.html** với ‘data’ = data đã lấy ở trên.



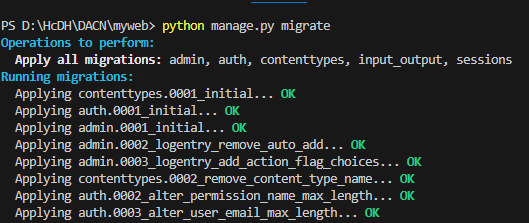
Cấu hình đường dẫn trong **urls.py**



Để có thể chạy được web có sử dụng cơ sở dữ liệu trong **Models** ta cần chạy lệnh **Python manage.py makemigrations** để Django tạo ra cơ sở dữ liệu.

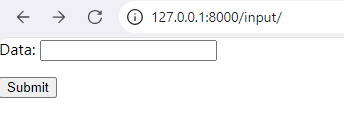


Tiếp theo đó, ta chạy lệnh **Python manage.py migrate** để cấu hình lại trang web cập nhật các thay đổi.

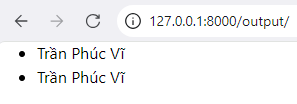


Sau khi đã cập nhật các thay đổi, ta chạy lệnh **Python manage.py runserver** để khởi động trang web, truy cập vào đường dẫn <http://127.0.0.1:8000/> để truy cập.

Đây là trang **input** khi nhập và ấn **Submit** dữ liệu sẽ được lưu vào cơ sở dữ liệu.

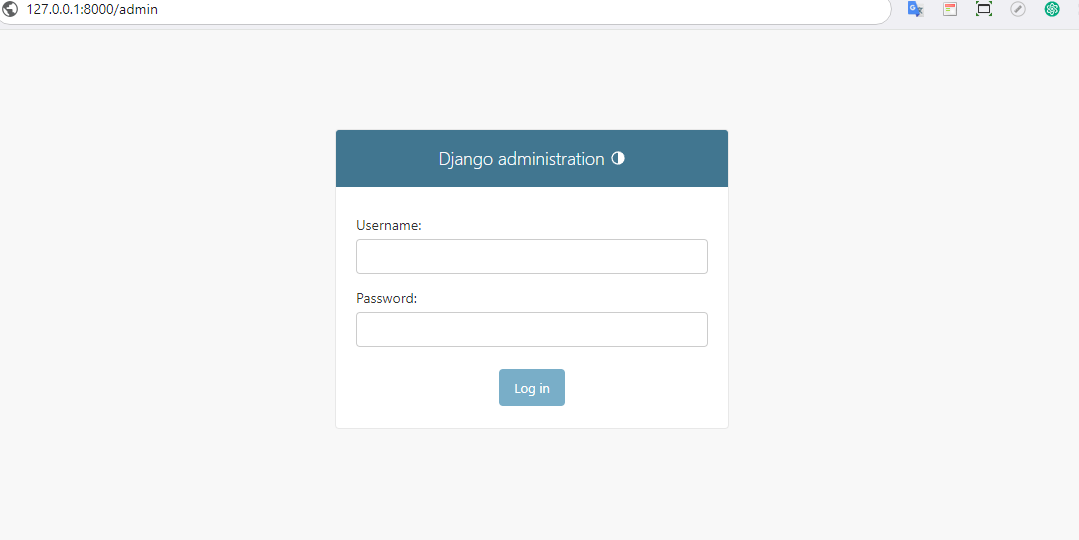


Đây là trang **output**, sau khi dữ liệu được nhập và ấn **Submit** sẽ tự động truy cập đến trang **output**, lấy tất cả dữ liệu và hiển thị lên thẻ <li></li>.



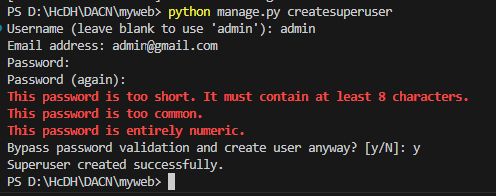
**4. Hệ thống admin**

Khi xây dựng một ứng dụng web như blog, cửa hàng trực tuyến, hay diễn đàn, không chỉ cần các trang hiển thị thông tin mà còn cần một trang admin để quản lý nội dung. Django cung cấp sẵn một trang quản trị mạnh mẽ, giúp người dùng quản lý nội dung dễ dàng bằng cách truy cập vào đường dẫn "/admin". Trang admin cung cấp nhiều tính năng quan trọng như thêm, sửa, xóa bài viết và cấu hình cho trang web. Điều này giúp người phát triển tiết kiệm thời gian và công sức trong việc quản lý nội dung của ứng dụng, để có thể truy cập vào trang admin, ta truy cập vào đường dẫn <http://127.0.0.1:8000/admin>.

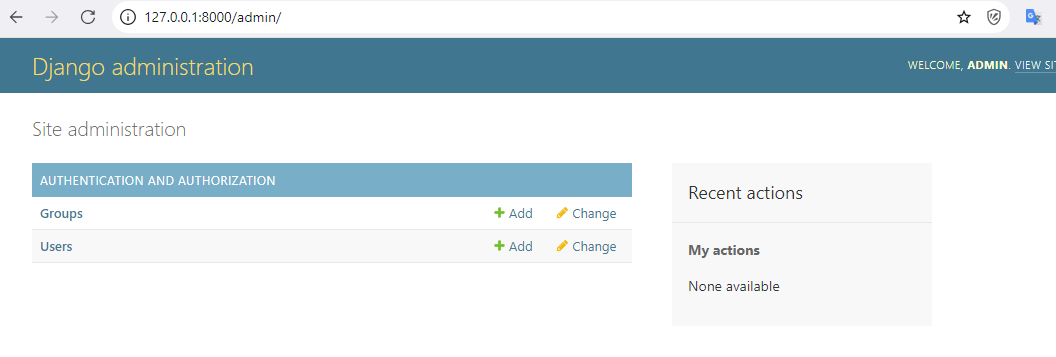


Nhưng để truy cập vào được bên trong của trang admin, ta cần phải có tài khoản admin. Để có thể tạo được tài khoản admin, ta sử dụng câu lệnh

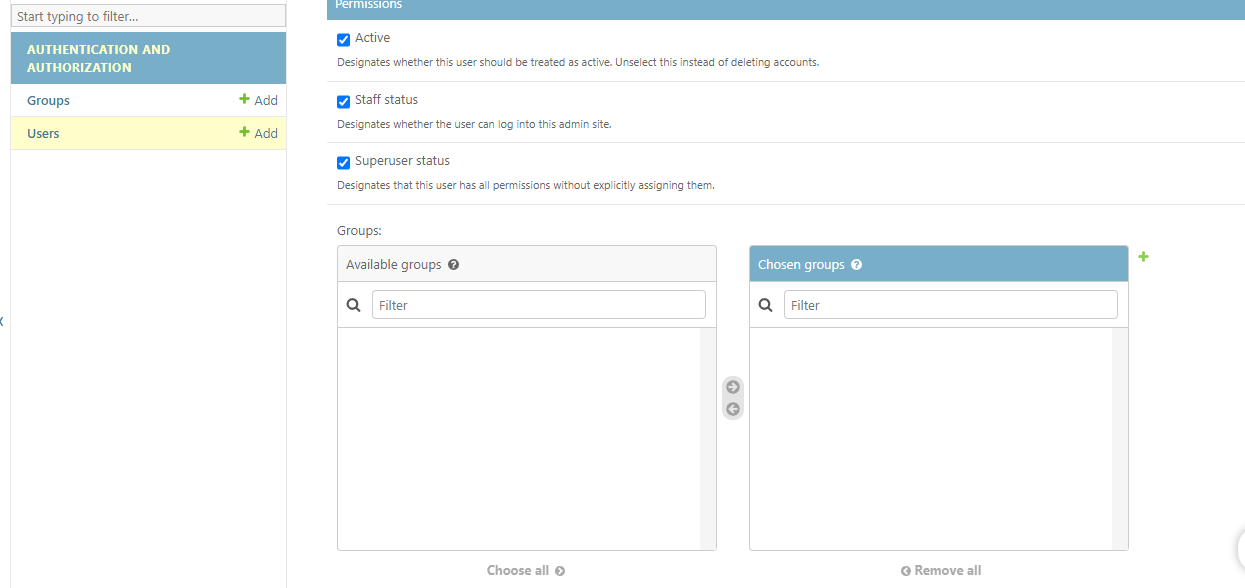
**Python manage.py createsuperuser**



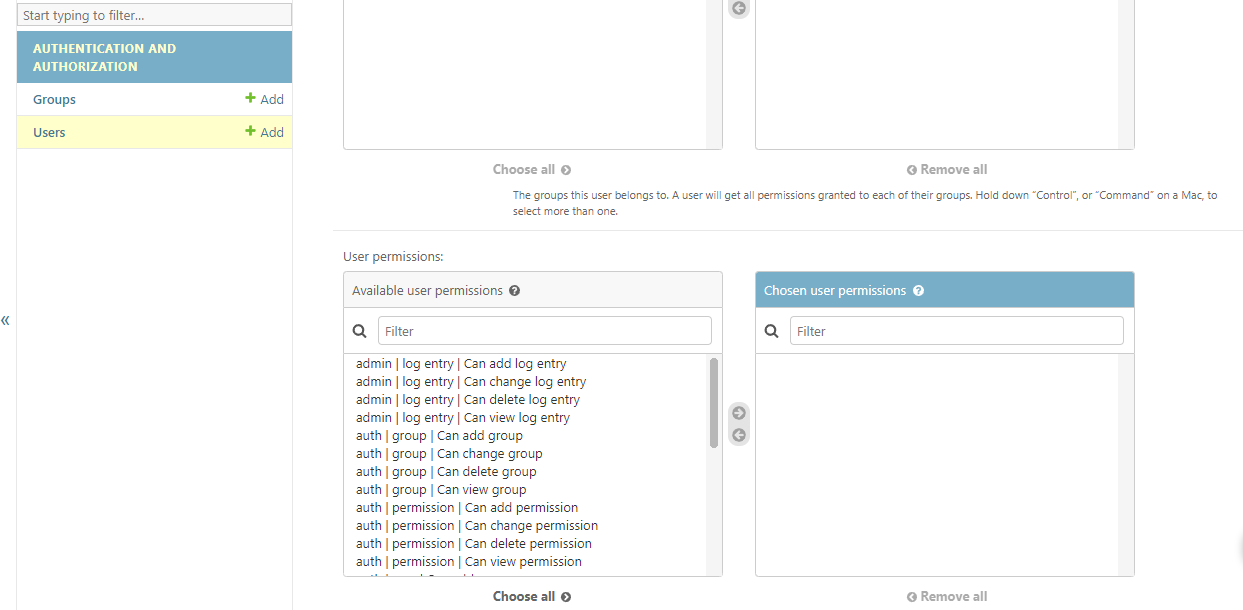
Khi đã tạo được tài khoản admin, ta tiến hành truy cập vào trang admin, theo đường dẫn trên, giao diện trang admin gồm: **Group** và **User**



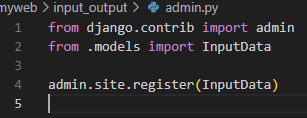
Trong trang admin, ta có thể thay thêm, sửa , xoá, phân quyền cho User có các quyền khác nhau gồm các quyền: có được phép truy cập vào trang admin hay không.



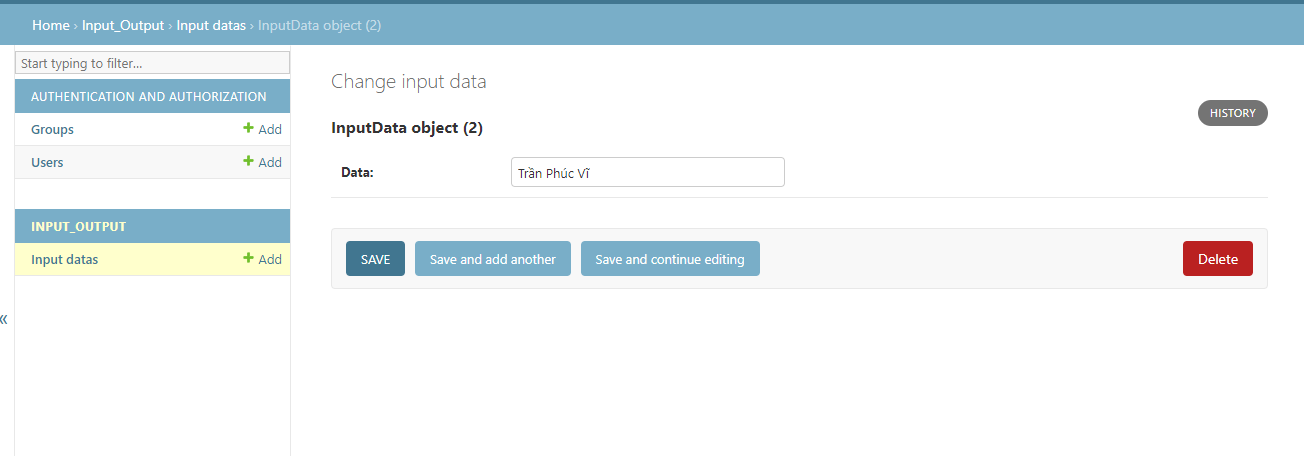
Còn ở đây dùng để cấp quyền cho User đó có các quyền gì đối với cơ sở dữ liệu trong hệ thông, như có thể xem, thêm, sửa, xoá Group,….



Và tại sao khi nảy có định nghĩa các lớp trong **Models** là các bảng trong cơ sở dữ liệu, vậy bản đó đang ở đâu. Để có thể truy cập được vào các bảng đã tạo trong **Models** ta cần phải **register** nó trong **admin.py,** đây là cấu trúc của phương thức **reigister**.



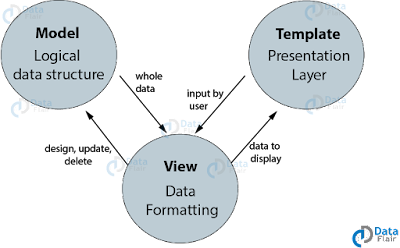
Sau khi đã đăng kí xong, thì bảng **Input datas** sẽ xuất hiện trên giao diện của admin, có thể truy cập vào để xem, thêm, xoá, sửa dữ liệu của bảng.



## 2.3 Mô hình MVT

### 2.3.1 Giới thiệu

MVT và MVC là hai mô hình phần mềm đều chia ứng dụng thành các phần: logic xử lý, hiển thị và tương tác cơ sở dữ liệu. MVT (Model-View-Template) là biến thể của MVC, nhưng chúng về cơ bản giống nhau.



**Model (Mô hình):** Đây là phần chịu trách nhiệm xử lý dữ liệu và logic của ứng dụng. Mô hình biểu diễn cấu trúc dữ liệu và các chức năng xử lý dữ liệu, bao gồm truy xuất, cập nhật và xử lý logic.

**View (Giao diện):** Phần này là nơi hiển thị thông tin cho người dùng. Giao diện là thành phần tương tác với người dùng và hiển thị dữ liệu từ mô hình. Trong MVT, Template (mẫu giao diện) giúp tạo ra giao diện từ dữ liệu được cung cấp bởi mô hình.

**Template (Mẫu):** Đây là thành phần tạo ra giao diện dựa trên dữ liệu từ mô hình. Nó có vai trò tương tự như View trong MVC, nhưng thường chịu trách nhiệm nhiều hơn về việc tạo ra cấu trúc của giao diện từ dữ liệu.

|  |  |  |
| --- | --- | --- |
| **Mô hình MVC** | **Mô hình MVT** | **Chức năng** |
| Model | Model | Làm việc với cơ sở dữ liệu |
| View | Template | Hiển thị giao diện |
| Controller | View | Xử lý logic |

### 2.3.2 Sự hoạt động của mô hình MVT

- Khi người dùng tương tác các hoạt động với hệ thống, yêu cầu sẽ được gửi đến Model.

- Model sẽ xử lý yêu cầu này và trả về dữ liệu cần thiết với yêu cầu

- Tiếp đó kết quả sẽ được Model gửi cho Templates, tại đây là nơi dữ liệu sẽ được chuyển thành giao diện.

- Cuối cùng, giao diện sẽ được hiển thị thông qua các trình duyệt theo yêu cầu của người dùng

### 2.3.3 Ưu nhược điểm của mô hình

**Ưu điểm:**

Nói về ưu điểm của mô hình MVT trước hết sẽ nói về khả năng quản lý và phát triển dự án, nhờ vào sự tách biệt của 3 phần (Model-View-Template), ta có thể dễ dàng quản lý cũng như kiểm lỗi và phát triển hệ thống.

Tiếp theo là khả năng linh hoạt, nhờ vào sự tách biệt mà mô hình MVT rất linh hoạt, khi việc có thể thay đổi một thành phần mà không ảnh hưởng nhiều đến các phần khác.

Việc có khả năng tách biệt 3 phần cũng góp phần giúp cho hiệu suất được nâng cao.

**Nhược điểm:**

Khả năng tách biệt tuy điểm lại nhiều ưu điểm, nhưng đó cũng tồn tại nhược điểm, việc tách biệt các thành phần yêu cầu khả năng hiểu biết chuyên sâu về cả 3 lĩnh vực (Model, View, Template) để có thể làm việc hiệu quả.

Việc đồng bộ và xử lý các logic cũng trở nên phước tạp và khó khăn hơn trong quá trình phát triển

# : HIỆN THỰC HÓA NGHIÊN CỨU

## 3.1 Mô tả bài toán

Để có thể xây dựng nên một web bán quần áo thể thao, trước hết ta cần nắm rõ tình hình của thị trường cũng như thị hiếu của khách hàng, xem xét các đối tượng nào sẽ là đối tượng tiềm năng trong tương lai.

Trước tiên ta cần phải khảo sát khắp các mạng xã hội cũng nhưng các trang web khác để có thể cập nhật tình hình chung, sau đó tiến hành thiết kế các cơ sở dữ liệu và tiến hành vào thực hiện dự án.

**Phân quyền cho hệ thống**

Bộ phận quản lý: là một trong những bộ phần có quyền hạn trong hệ thống, chịu trách nhiệm quản lý tài khoản của khác hàng, cũng như xem xét cập nhật các mặt hàng mới trên thị trường, khảo sát tình hình, cập nhật lại thông tin, có khả năng thêm mới các sản phẩm, cũng như tạo mới tài khoản cho người dùng, có thể xem thông tin của các sản phẩm, có quyền sửa sản phầm, nhưng sẽ không có quyền xoá sản phầm.

Bộ phận quản trị: Có vai trò cao nhất trong hệ thống, đảm nhiệm chức năng tạo mới các quản lý, thêm xoá các quản lý và cũng có toàn quyền đối với hệ thống, có khả năng thêm, sửa, xoá các loại sản phẩm.

**Quy trình làm việc**

Khách hàng sẽ truy cập vào website thông qua địa chỉ URL, hoặc vào các công cụ tìm kiếm của trình duyệt để tìm kiếm.

Sau khi đã truy cập được vào website, khách hàng có thể xem xét các sản phẩm mà mình yêu thích, có thể xem các sản phẩm theo từng loại khác nhau được phân chia cụ thể rõ ràng.

Nếu khách hàng đã có sản phẩm muốn mua từ trước thì có thể tìm kiếm sản phẩm đó trên thanh tìm kiếm của website, nếu hệ thống có tồn tại sản phẩm thì sẽ trả về cho khách hàng các thông tin của sản phẩm cũng như các mặt hàng có tên tương tự.

Sau khi đã tìm thấy sản phẩm, khách hàng có thể vào xem các thông tin chi tiết của sản phẩm đã được cung cấp bên trong, khi đã ưng ý khách hàng có thể ấn mua để thêm vào giỏ hàng, sau đó khách hàng có thể mua các sản phẩm khác hoặc có thể tiến hành tới bước xác nhận.

Khi đã chọn xong các sản phẩm ưng ý, khách hàng sẽ vào giỏ hàng để chỉnh sửa lại số lượng các sản phẩm mà khách hàng đã chọn, thêm các thông tin chi tiết như: họ tên, địa chỉ, số điện thoại,... và tiến hành xác nhận để hoàn tất.

Sau khi khách hàng đã xác nhận, trang quản trị sẽ được cập nhật đơn hàng, và tiến hàng chuẩn bị hàng giao cho khách.

**Yêu cầu về chức năng**

**-** Trang chủ: tổng quan về nội dung của website, thông tin liên lạc cũng như địa chỉ của website.

- Trang sản phẩm: thể hiện toàn bộ sản phẩm hiện có trong hệ thống, danh mục sản phẩm,…

- Trang thông tin sản phẩm: thể hiện cụ thể tên của sản phầm, giá cả, cũng nhưng các thông tin chi tiết giúp khách hàng nắm rõ được chi tiết của sản phẩm, ngoài ra có thể có các hình từ nhiều góc khác nhau của sản phẩm để khách hàng tham khảo.

- Giỏ hàng: lưu trữ được các sản phẩm khách hàng đã từng đưa vào giỏ hàng, có khả năng thay đổi số lượng sản phẩm theo khách hàng mong muốn.

- Tìm kiếm: hiển thị ra được các sản phẩm mà khách hàng tìm kiếm theo tiêu chí là tên sản phẩm.

**Yêu cầu phi vật lí:**

- Để nói về một website bán hàng online, đầu tiên phải nói đến giao diện, giao diện phải ưu nhìn, trình bài rõ ràng, màu sắc hài hoà, không để mọi thứ quá lộn xộn, đảm bảo tính thẩm mỹ, tạo ấn tượng tốt với người dùng khi vừa truy cập vào website.

- Các chức năng phải dễ dàng sử dụng, phù hợp với hầu hết các độ tuổi của khách hàng, không nên quá phức tạp sẽ gây khó khăn cho một vài độ tuổi khách hàng.

## 3.2 Mô tả bài toán

### 3.2.1 Mô hình Use Case tổng quát:

