# Chapter 1: Initial Set Up

## The Command Line, Install Python 3 Windows

phiên bản gần đây nhất của Django (2.0), Python (3.7) và Pipenv để quản lý môi trường ảo

python –version

## Virtual Environments, Install Django

Bạn nên sử dụng môi trường ảo chuyên dụng cho mỗi dự án Python mới.

Trong lịch sử, các nhà phát triển Python đã sử dụng virtualenv hoặc pyenv để định cấu hình môi trường ảo. Nhưng vào năm 2017, nhà phát triển Python nổi tiếng Kenneth Reitz đã phát hành Pipenv hiện là công cụ đóng gói Python được khuyến nghị chính thức .

pip3 install pipenv

sau đó :

cd ~/Desktop

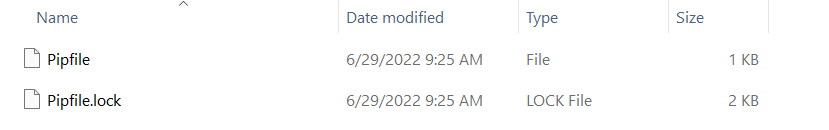
mkdir django

cd django

và :

pipenv install django==2.1

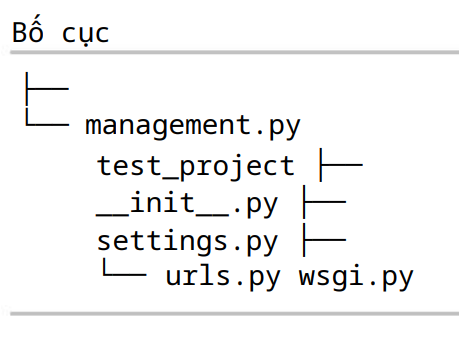
sẽ tạo ra :



pipenv shell

django-admin startproject test\_project .

Bằng cách chạy django admin startproject test\_project. Với dấu chấm ở cuối – có nội dung là cài đặt trong thư mục hiện tại – thay vào đó kết quả là:



python manage.py runserver

đánh trên cốc cốc :

<http://127.0.0.1:8000/>

exit / ctrl C

## Install Git

[Git for Windows](https://gitforwindows.org/)

git config --global user.name "Your Name"

git config --global user.email "yourname@email.com"

## Text Editors

hãy tải xuống và cài đặt Visual Studio Code bây giờ: [Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/)

# Chapter 2: Hello World app

## Initial Set Up

cd ~/Desktop

mkdir helloworld

cd helloworld

pipenv install django==2.1

pipenv shell

django-admin startproject helloworld\_project .

python manage.py runserver

## Create an app

Ctrl C

python manage.py startapp pages

ta thấy có các thông số :

admin.py là một tệp cấu hình cho ứng dụng Quản trị Django được tích hợp sẵn

apps.py là một tệp cấu hình cho chính ứng dụng

migrations: theo dõi bất kỳ thay đổi nào đối với tệp models.py để framework là một kỹ thuật nâng cao. Đối với ứng dụng tương đối cơ bản của chúng tôi cơ sở dữ liệu và model.py luôn đồng bộ hóa

models.py là nơi chúng tôi xác định các mô hình cơ sở dữ liệu của mình, mà Django tự động dịch thành các bảng cơ sở dữ liệu

tests.py dành cho các thử nghiệm dành riêng cho ứng dụng của chúng tôi

views.py là nơi chúng tôi xử lý logic yêu cầu / phản hồi cho ứng dụng web của chúng tôi

## Views and URLConfs

Vào xem ở setting

Vào pages/view.py thêm mới

Thêm 1 fule url :

npm install touch-cli –g

touch pages/urls.py ( tại PS C:\Users\huyre\Desktop\helloworld> touch pages/urls.py)

Cập nhật tệp helloworld\_project / urls.py

## Hello, world!

python manage.py runserver

# Chapter 3: Pages app

## Initial Set Up

As in Chapter 2: Hello World app, our initial set up involves the following steps:

create a new directory for our code

install Django in a new virtual environment

create a new Django project

create a new pages

app update settings.py

## Templates

mkdir templates

touch templates/home.html

sau đó update settings.py

'DIRS': [os.path.join(BASE\_DIR, 'templates')], # new

Sau đó Update the pages/views.py file. + url.py file

# Chapter 4: Message Board app

## Initial Set Up

pipenv install django==4.0

pipenv shell

python manage.py migrate

python manage.py runserver

## Create a database model

Update # posts/models.py

## Activating models

python manage.py makemigrations posts

python manage.py migrate posts

## Django Admin

python manage.py createsuperuser

Username : huyinit

Email:

Password: huyinit

Password : huyinit

Superuser created successfully

Sau đó vào :

<http://127.0.0.1:8000/admin/>

nếu bị lỗi :

python manage.py makemigrations

python manage.py migrate

python manage.py migrate --run-syncdb

Thấy chưa có main => update # posts/admin.py

## Views/Templates/URLs

Update # posts/views.py

mkdir templates

touch templates/home.htm

update # posts/views.py

## Adding new posts

<http://127.0.0.1:8000/admin/>

save lại

vào mở lại http://127.0.0.1:8000/

# Chapter 5: Blog app

## Initial Set Up

cd ~/Desktop

mkdir blog

cd blog

pipenv install django==4.0

pipenv shell

django-admin startproject blog\_project .

python manage.py startapp blog

python manage.py migrate

python manage.py runserver

vào setting.py :

'blog.apps.BlogConfig', # new

## Database Models

Viết model

Update # blog/admin.py : admin.site.register(Post)

Sau đó :

python manage.py makemigrations blog

python manage.py migrate blog

## Admin

python manage.py createsuperuser

111 và 111 nhé

## Views

Update # blog/views.py

## URLs

touch blog/urls.py

update # blog/urls.py

=>update # blog\_project/urls.py

## Static files

mkdir static

mkdir static/css

touch static/css/base.css

=>update : # blog\_project/settings.py (STATICFILES\_DIRS = [os.path.join(BASE\_DIR, 'static')])

And update templates/base.html

## Templates

mkdir templates

touch templates/base.html

touch templates/home.html

sau update : # blog\_project/settings.py

update : templates/base.html + templates/home.html

## Individual blog pages

Update : # blog/views.py

touch templates/post\_detail.html

update templates/post\_detail.html

=>update # blog/urls.py

Update templates/home.html

# Chapter 6: Forms

## Forms

pipenv install django==4.0

pipenv shell

python manage.py migrate

python manage.py runserver

Update templates/base.html

Thêm link trong # blog/urls.py

Update hàm # blog/views.py

touch templates/post\_new.html

http://127.0.0.1:8000/post/new/

Chúng tôi phàn nàn rằng chúng tôi không chỉ định nơi gửi người dùng sau khi gửi biểu mẫu thành công. thêm get\_absolute\_url . Nó đặt URL chuẩn cho một đối tượng, vì vậy, ngay cả khi cấu trúc URL của bạn thay đổi trong tương lai, thì tham chiếu đến đối tượng cụ thể vẫn giống nhau

## Update Form (edit)

Update templates / post\_detail.html :< a href="{% url'post\_edit' post.pk %}">

touch templates/post\_edit.html

edit post\_edit.html

update #blog / views.py khi ấn nút cập nhật

update # blog/urls.py

## Delete View

Update templates/post\_detail.html

touch templates/post\_delete.html

update # blog/views.py

update # blog/urls.py

# Chapter 7: User Accounts

## Log in

Update # blog\_project/urls.py : path('accounts/', include('django.contrib.auth.urls')), # new

mkdir templates/registration

touch templates/registration/login.html

# blog\_project/settings.py :

LOGIN\_REDIRECT\_URL = 'home' + LOGOUT\_REDIRECT\_URL = 'home' # new

Python manage.py runserver

<http://127.0.0.1:8000/accounts/login/>

## Updated homepage

Update templates/base.html

## Sign up

python manage.py startapp accounts

Update # blog\_project/settings.py : 'accounts.apps.AccountsConfig', # new

Update blog\_project/urls.py : path('accounts/', include('accounts.urls')), # new

touch accounts/urls.py

update # acounts/urls.py

# accounts/views.py

touch templates/registration/signup.html

update templates/ registration/signup.html

<http://127.0.0.1:8000/accounts/signup/>

: Signup -> Login -> Homepage

# Chapter 8: Custom User Model

## Set Up Custom

cd ~/Desktop

mkdir chapter8\_custom\_user\_model

cd chapter8\_custom\_user\_model

pipenv install django==4.0

pipenv shell

django-admin startproject newspaper\_project .

python manage.py startapp users

python manage.py runserver

## Set Up Custom

# newspaper\_project/settings.py

có

'users.apps.UsersConfig',  # new

AUTH\_USER\_MODEL = 'users.CustomUser' # new

sử dụng cấu hình AUTH\_USER\_MODEL để yêu cầu Django sử dụng mô hình người dùng tùy chỉnh mới của chúng tôi thay cho mô hình Người dùng tích hợp

# users/models.py

một mô hình Người dùng mới mà chúng tôi sẽ gọi là CustomUser mở rộng AbstractUser hiện có

## Forms

hai cách mà chúng ta sẽ tương tác với mô hình CustomUser mới của mình là gì? Một trường hợp là khi người dùng đăng ký tài khoản mới trên trang web của chúng tôi. Ứng dụng còn lại nằm trong ứng dụng quản trị , cho phép chúng tôi, với tư cách là người quản lý cấp cao, sửa đổi người dùng hiện có.

=> UserCreationForm and UserChangeForm.

touch users/forms.py

# users/forms.py

sử dụng các trường mặc định thông qua Meta.fields bao gồm tất cả các trường mặc định.

# users/admin.py

python manage.py makemigrations users

python manage.py migrate

## Superuser

python manage.py createsuperuser

python manage.py runserver

<http://127.0.0.1:8000/admin>

click on the link for “Users”

tại lớp CustomUserAdmin : list\_display = ['email', 'username', 'age', 'is\_staff', ] # new

# Chapter 9: User Authentication

## Templates

mkdir templates

mkdir templates/registration

# newspaper\_project/settings.py:

'DIRS': [os.path.join(BASE\_DIR, 'templates')], # new

LOGIN\_REDIRECT\_URL = 'home' LOGOUT\_REDIRECT\_URL = 'home'

touch templates/registration/login.html

touch templates/base.html

touch templates/home.html

touch templates/registration/signup.html

## URLs

touch users/urls.py

# users/urls.py

# newspaper\_project.urls

# users/views.py

## Admin

python manage.py runserver

<http://127.0.0.1:8000/admin>

<http://127.0.0.1:8000/users/signup/>

=># users/forms.py

# Chapter 10: Bootstrap

## Pages app

python manage.py startapp pages

# newspaper\_project/settings.py: 'pages.apps.PagesConfig', # new

# newspaper\_project/urls.py: path('', include('pages.urls')), # new

touch pages/urls.py

# pages/urls.py

# pages/views.py

Tests

python manage.py test

## Bootstrap

pip install crispy-bootstrap5

!-- templates/base.html

!-- templates/registration/login.html –

## Sign Up Form

Làm việc với các biểu mẫu là một thách thức và các biểu mẫu django-giòn giúp viết mã KHÔ dễ dàng hơn

pipenv install django-crispy-forms==1.7.2

# newspaper\_project/settings.py: 'crispy\_forms', # new

Và: CRISPY\_TEMPLATE\_PACK = 'bootstrap4'

pip install crispy-bootstrap5

-- templates/signup.html -->

python manage.py runserver

# Chapter 11: Password Change and Reset

pipenv install django==4.0

pipenv shell

pipenv install django-crispy-forms==1.7.2

pip install crispy-bootstrap5

python manage.py runserver

## Password Change

<http://127.0.0.1:8000/users/password_change/>

## Customizing password change

touch templates/registration/password\_change\_form.html

touch templates/registration/password\_change\_done.html

## Password reset

chúng tôi sẽ sử dụng dịch vụ email SendGrid để thực sự gửi email nhưng với mục đích thử nghiệm, chúng tôi có thể dựa vào phần phụ trợ bảng điều khiển của Django thay vào đó, cài đặt xuất văn bản email tới bảng điều khiển dòng lệnh của chúng tôi.

# newspaper\_project/settings.py

EMAIL\_BACKEND = 'django.core.mail.backends.console.EmailBackend'

http://127.0.0.1:8000/users/password\_reset/

http://127.0.0.1:8000/users/password\_reset/done/

## Custom Templates

touch templates/registration/password\_reset\_form.html

touch templates/registration/password\_reset\_done.html

touch templates/registration/password\_reset\_confirm.html

touch templates/registration/password\_reset\_complete.html

http://127.0.0.1:8000/users/password\_reset/

<https://support.sendgrid.com/hc/en-us/requests/new/account-access-issue>

<https://signup.sendgrid.com/declined>

# Chapter 13: Newspaper app

pipenv install django==4.0

pipenv shell

pipenv install django-crispy-forms==1.7.2

pip install crispy-bootstrap5

python manage.py runserver

## Articles app

python manage.py startapp articles

# newspaper\_project/settings.py:

'articles.apps.ArticlesConfig', # new

TIME\_ZONE = 'Asia/Ho\_Chi\_Minh'

# articles/models.py

python manage.py makemigrations articles

python manage.py migrate

# articles/admin.py

python manage.py runserver

## URLs and Views

# newspaper\_project/urls.py

touch articles/urls.py

# articles/urls.py

# articles/views.py

touch templates/article\_list.html

!-- templates/article\_list.html

## Edit/Delete

# articles/urls.py

# articles/views.py

touch templates/article\_detail.html

touch templates/article\_edit.html

touch templates/article\_delete.html

edit cả : !-- templates/article\_list.html

## Create page

# articles/views.py

# articles/urls.py

touch templates/article\_new.html

!-- templates/article\_new.html –

!-- templates/base.html -->

!-- templates/home.html -->

Conclusion

# Chapter 14: Permissions and Authorization

## Improved CreateView

# articles/views.py

Sau đó xem lại giao diện khi new article

## Authorizations

http://127.0.0.1:8000/articles/new/?

Nhưng lỗi do thiếu trường tác giả

## Mixins

# articles/views.py

from django.contrib.auth.mixins import LoginRequiredMixin # new

class ArticleCreateView(LoginRequiredMixin, CreateView): # new

http://127.0.0.1:8000/articles/new/

= 'login' # new

## LoginRequiredMixin

login\_url = 'login' # new

## UpdateView and DeleteView

# articles/views.py

UserPassesTestMixin # new

def test\_func(self): # new

python manage.py runserver

Conclusion

# Chapter 15: Comments

Model

Admin

Template

Conclusion

Conclusion

# GITHUB và heroku

git config --global user.name "huyreeve"

git config --global user.email “huyreeve@gmail.com”

ssh-keygen

C:\Users\huyre\.ssh

git init

git status

git add -a

git commit -m 'initial commit'

git pull origin master

Bitbucket

git remote add origin [git@bitbucket.org:/hello-world.git](mailto:git@bitbucket.org:/hello-world.git)

git remote add origin git@bitbucket.org:wanghiinit/kkk.git

git push -u origin master

Your git URL might have changed. Change the URL in the local directory by using the following command

for **https** protocol

git remote set-url origin https://github.com/username/repository.git

for **ssh** protocol

git remote set-url origin git@github.com:username/repository.git

exit

Heroku

<https://dashboard.heroku.com/apps>

The Heroku CLI

[The Heroku CLI | Heroku Dev Center](https://devcenter.heroku.com/articles/heroku-cli#download-and-install)

heroku login

Additional Files

update Pipfile.lock make a new Procfile file install gunicorn as our web server make aone-line change to settings.py file

pipenv lock

touch Procfile

=> web: gunicorn pages\_project.wsgi --log-file –

pipenv install gunicorn==19.9.0

# pages\_project/settings.py => ALLOWED\_HOSTS = ['\*']

git status

git add –A

git commit -m "New updates for Heroku deployment"

git push -u origin master

Deploy

heroku create

heroku git:remote -a fathomless-hamlet-26076

heroku config:set DISABLE\_COLLECTSTATIC=1

git push heroku master

heroku ps:scale web=1

heroku open

………user

heroku login

heroku create dfb-blog

heroku git:remote -a dfb-blog

pipenv install whitenoise==3.3.1

# blog\_project/settings.py

git add –A

git commit -m 'Heroku config'

git push origin master

git push heroku master

heroku ps:scale web=1

# [ch16-deployment](https://github.com/wsvincent/djangoforbeginners/tree/master/ch16-deployment)

pip install -r requirements.txt

pipenv install django==4.0

pipenv shell

pipenv install django-crispy-forms==1.7.2

pip install crispy-bootstrap5

python manage.py runserver

python manage.py makemigrations

python manage.py migrate