Python網站框架絕技: Django 完全攻略班

廣宣學堂

吳柏翰 (Jerry Wu)

簡介

- 學歷
  - 台科大資管博士候選人
- 現任
  - 亞太智能機器 Founder
  - 國立台灣科技大學 講師
  - MIT edX/MandarinX 首席軟體工程師
- 經歷
  - 裕隆集團/華創車電 資深數據科學顧問
  - DSP智庫驅動 共同創辦人兼技術長
  - 美商Teradata 數據科學顧問
  - 精誠集團 數據科學顧問
  - 中華R軟體學會創會理事
  - 堂朝數位整合 資料分析師
- 網站
  - https://www.ap-mic.com
  - http://dataology.blogspot.tw
  - https://www.linkedin.com/in/jerry2012/

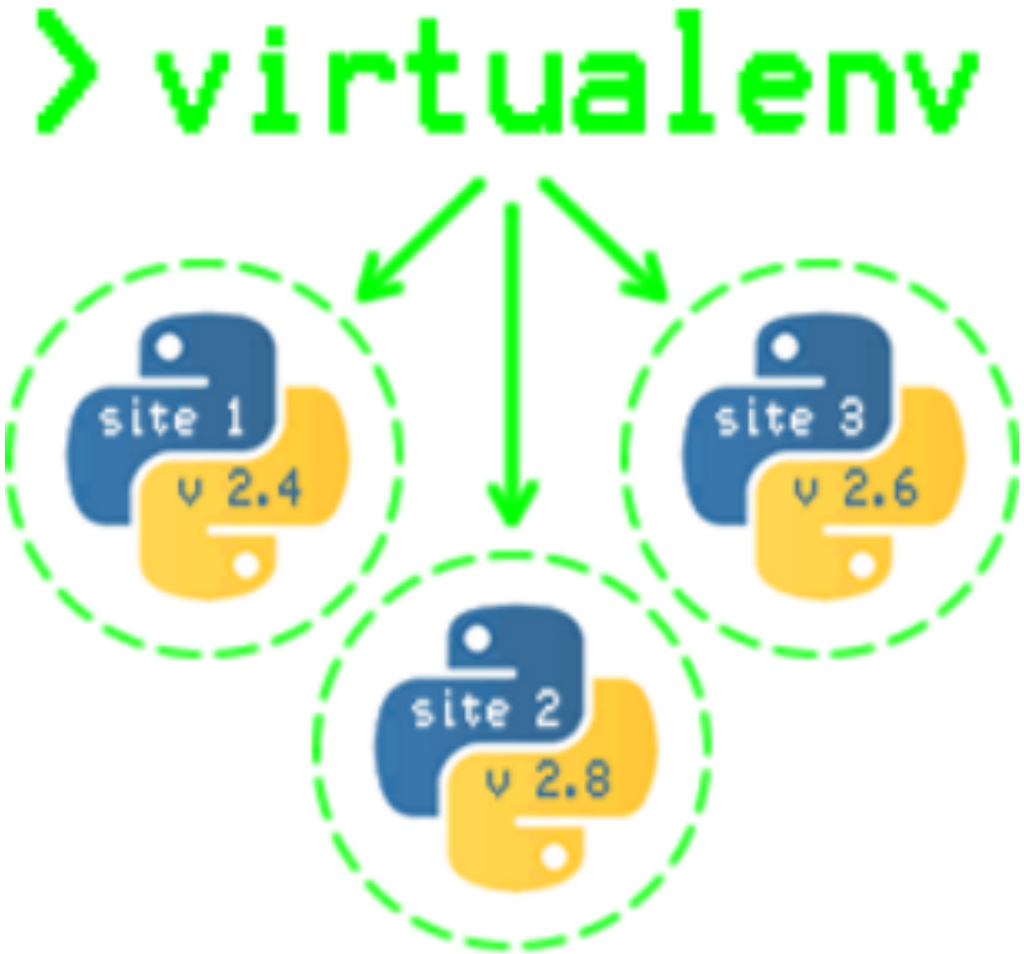
吳柏翰(Jerry)

Top 5 Python Web Framework

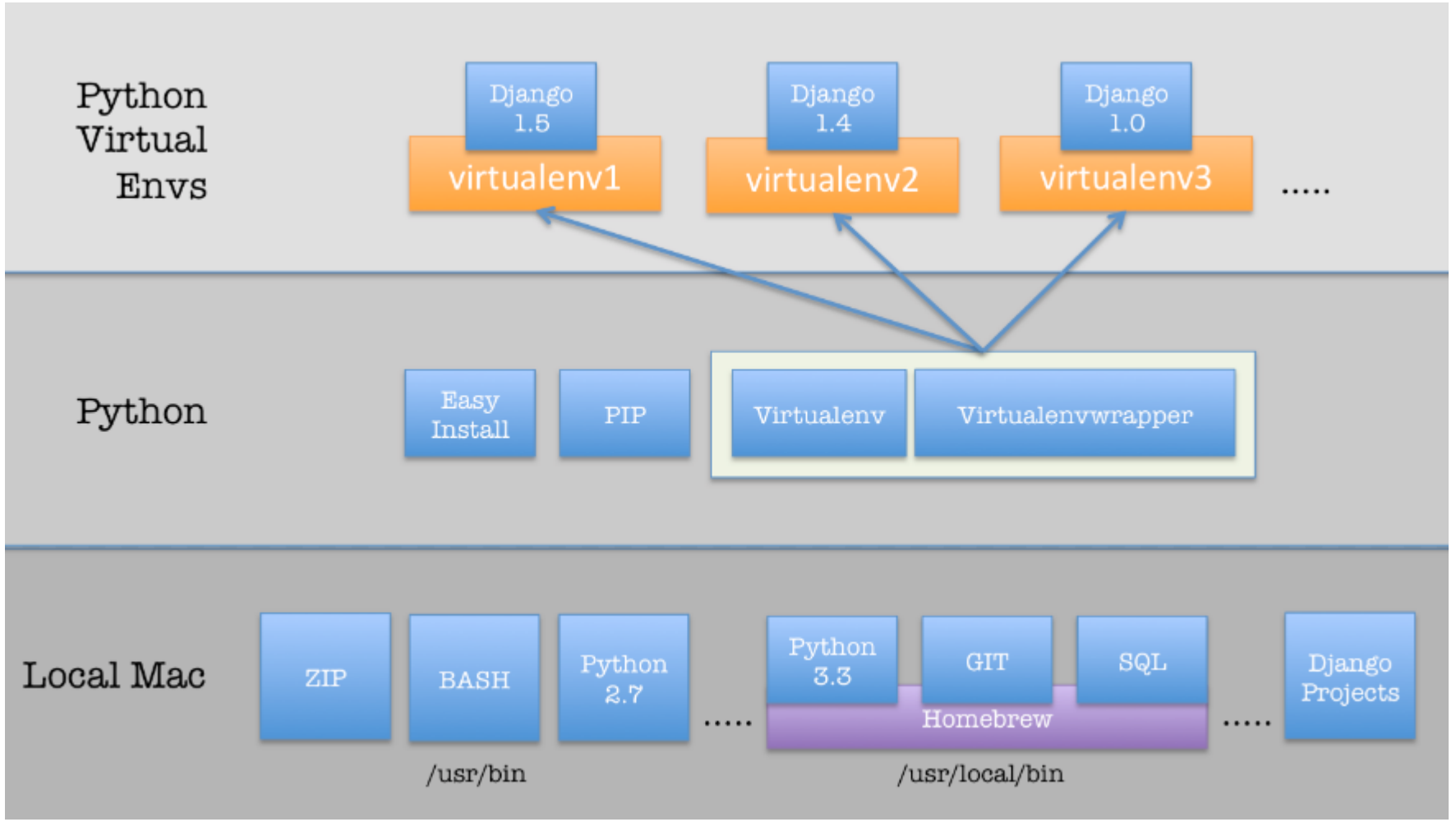
Joomla VS WP = Django VS Flask

- Django
  - 目前Python主流框架，適合新創團隊使用，論壇資料多，適合快速打造應用。
  - 框架完整是優點也是缺點
  - 知名案例：Instagram、edx、Pinterest
- Flask
  - 微框架，透過擴展來提升框架完整度，彈性高是優點也是缺點
  - 適合有開發經驗的團隊使用。案例：豆瓣小組
- Tornado
  - 異步非阻塞框架，適合用在需要響應非常快速的網站應用。
    - 同步：一件一件做，異步：每一件事情輪流做
      阻塞：一直等回覆，非阻塞：一直問
- Bottle
  - 微框架，適合開發小型網站，不適合商業應用。
- Web2py
  - 微框架，是由google在web.py二次開發，擴展性不佳。

Vitrualenv



Vitrualenv


Python Virtual Envs
Django 1.5
virtualenv1
Django 1.4
virtualenv2
Django 1.0
virtualenv3
.....
Python
Easy Install
PIP
Virtualenv
Virtualenvwrapper
Local Mac
ZIP
BASH
Python 2.7
.....
Python 3.3
GIT
SQL
Homebrew
.....
Django Projects
/usr/bin
/usr/local/bin

Vitrualenv

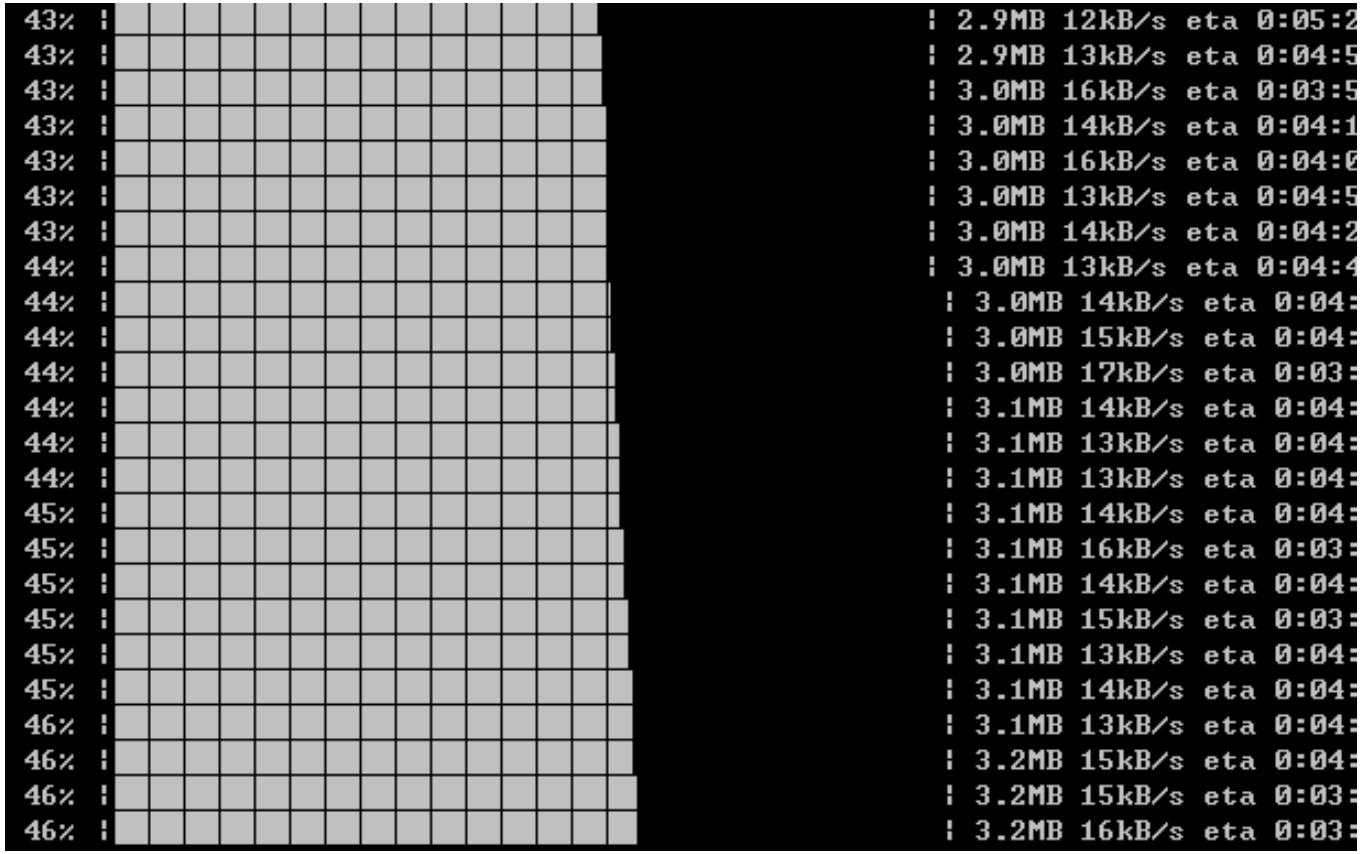
- pip install virtualenv
- pip3 install virtualenv
- easy_install virtualenv

- virtualenv
- virtualenv project

- WIN Scripts\activate.bat
- MAC
  - source ./bin/activate
  - Deactivate

Install Django

pip install django==1.11rc1



Install Django - Verifying

python -m django --version

```
(mywebsite) c:\GitHub\Django_new\mysite>python -m django --version
1.11rc1
```

Part 1: Requests and responses

Creating a project

- django-admin startproject mysite
- python manage.py runserver

```
mysite/
    manage.py
    mysite/
        __init__.py
        settings.py
        urls.py
        wsgi.py
```

Creating a project

- manage.py: A command-line utility that lets you interact with this Django project in various ways.
- mysite/__init__.py: An empty file that tells Python that this directory should be considered a Python package.
- mysite/settings.py: Settings/configuration for this Django project.
- mysite/urls.py: The URL declarations for this Django project.
- mysite/wsgi.py: An entry-point for WSGI-compatible web servers to serve your project.

Creating app

- What's the difference between a project and an app ?

- An app is a Web application that does something – e.g., a Weblog system, a database of public records or a simple poll app.

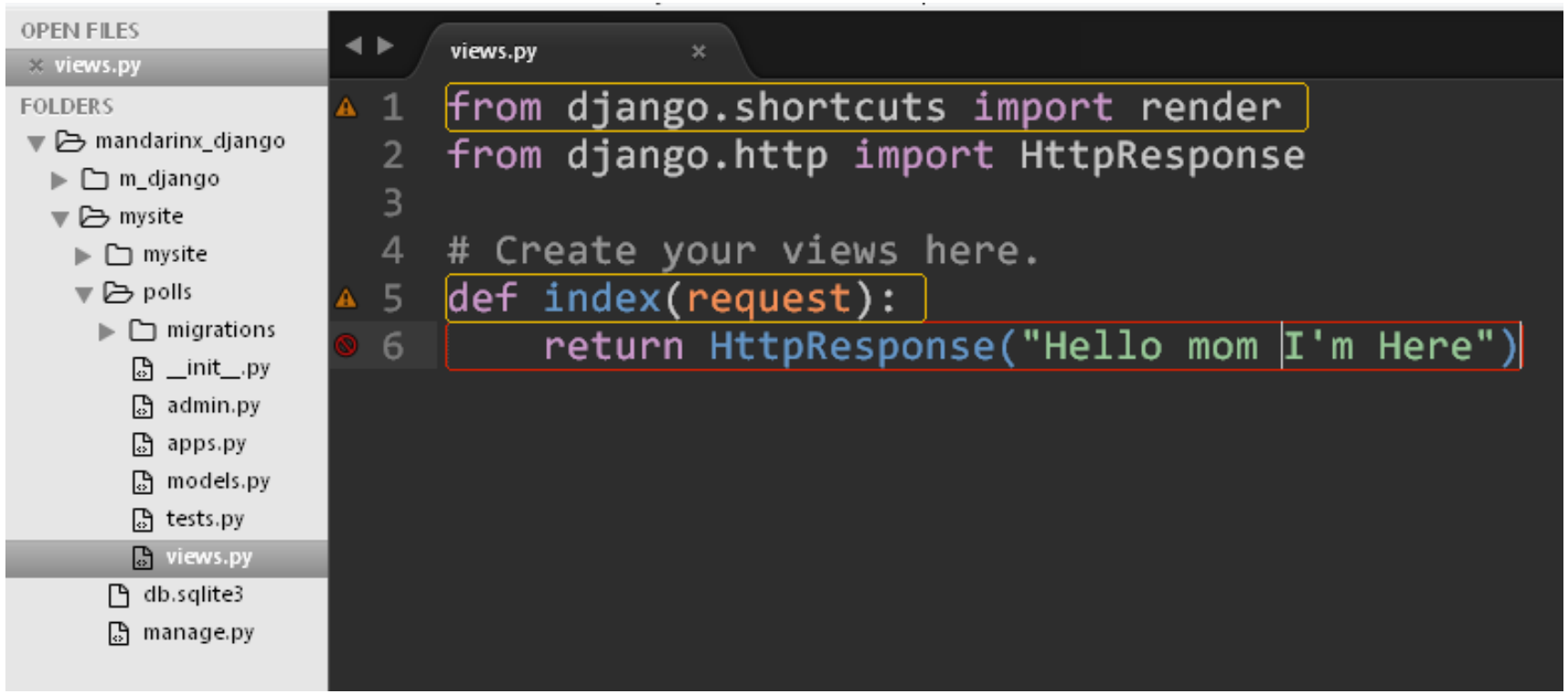
- A project is a collection of configuration and apps for a particular website. A project can contain multiple apps. An app can be in multiple projects.

Creating app

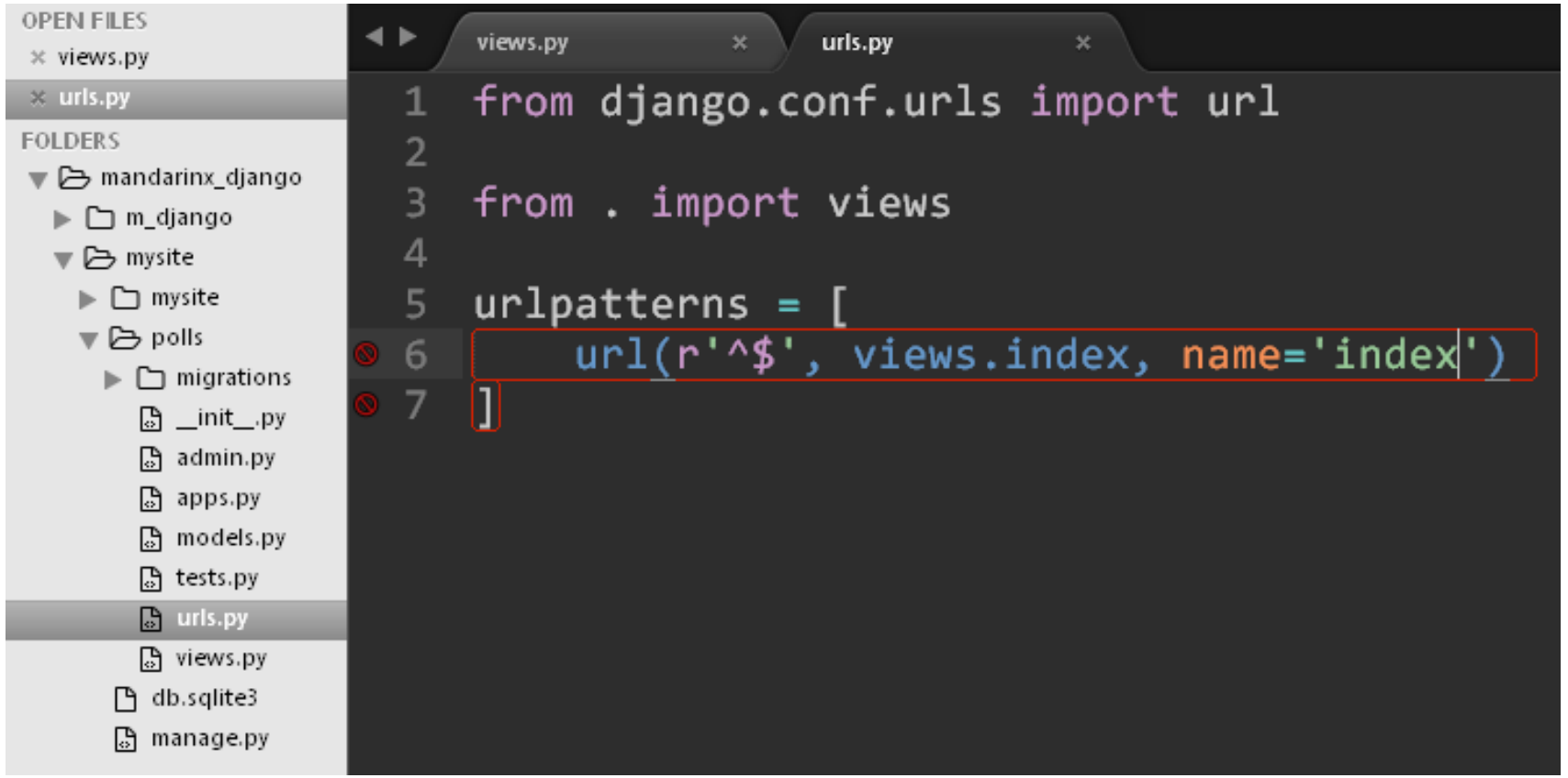
- python manage.py startapp jerry

```
polls/
    __init__.py
    admin.py
    apps.py
    migrations/
        __init__.py
    models.py
    tests.py
    views.py
```

Write your first view



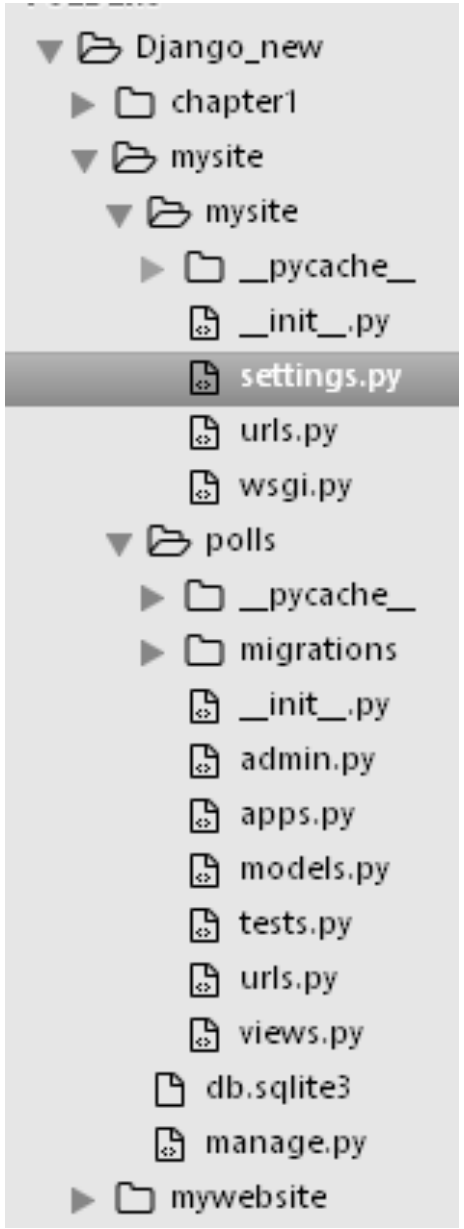
Write your first view



Write your first view

```
3  The `urlpatterns` list routes URLs to views. For more information please see:
4      https://docs.djangoproject.com/en/dev/topics/http/urls/
5  Examples:
6  Function views
7      1. Add an import:  from my_app import views
8      2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
9  Class-based views
10     1. Add an import:  from other_app.views import Home
11     2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
12 Including another URLconf
13     1. Import the include() function: from django.conf.urls import url, include
14     2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
15 """
16 from django.conf.urls import include, url
17 from django.contrib import admin
18
19 urlpatterns = [
20     url(r'^polls/', include('polls.urls')),
21     url(r'^admin/', admin.site.urls),
22 ]
23
```

dsjerry2017@gmail.com



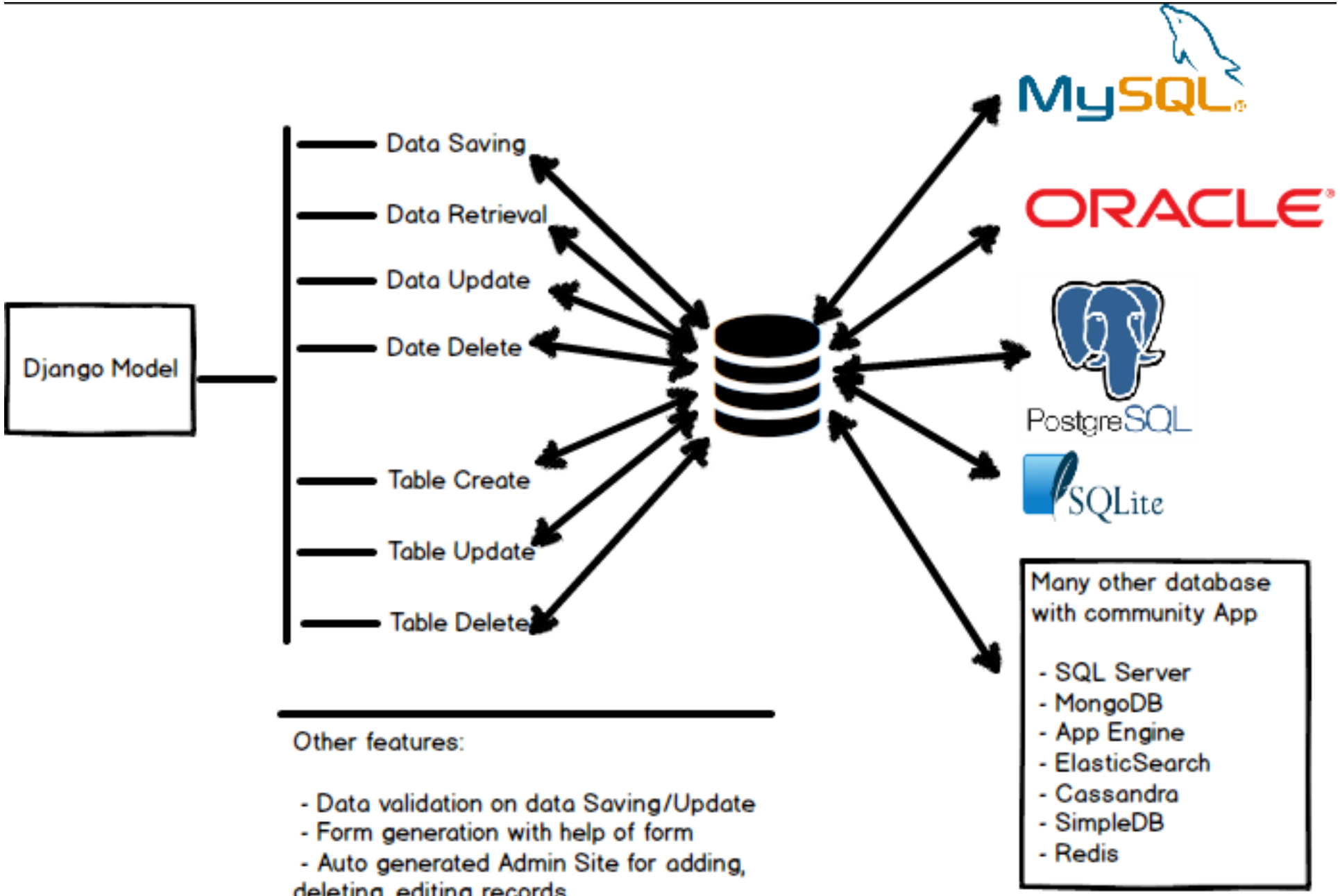

```python
INSTALLED_APPS = [
    'polls.apps.PollsConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

# Internationalization
# https://docs.djangoproject.com/en/dev/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Asia/Taipei'

USE_I18N = True

USE_L10N = True

USE_TZ = True
```

Part 2: Models and the admin site

Part 2: Models and the admin site

```
$ python manage.py migrate
```

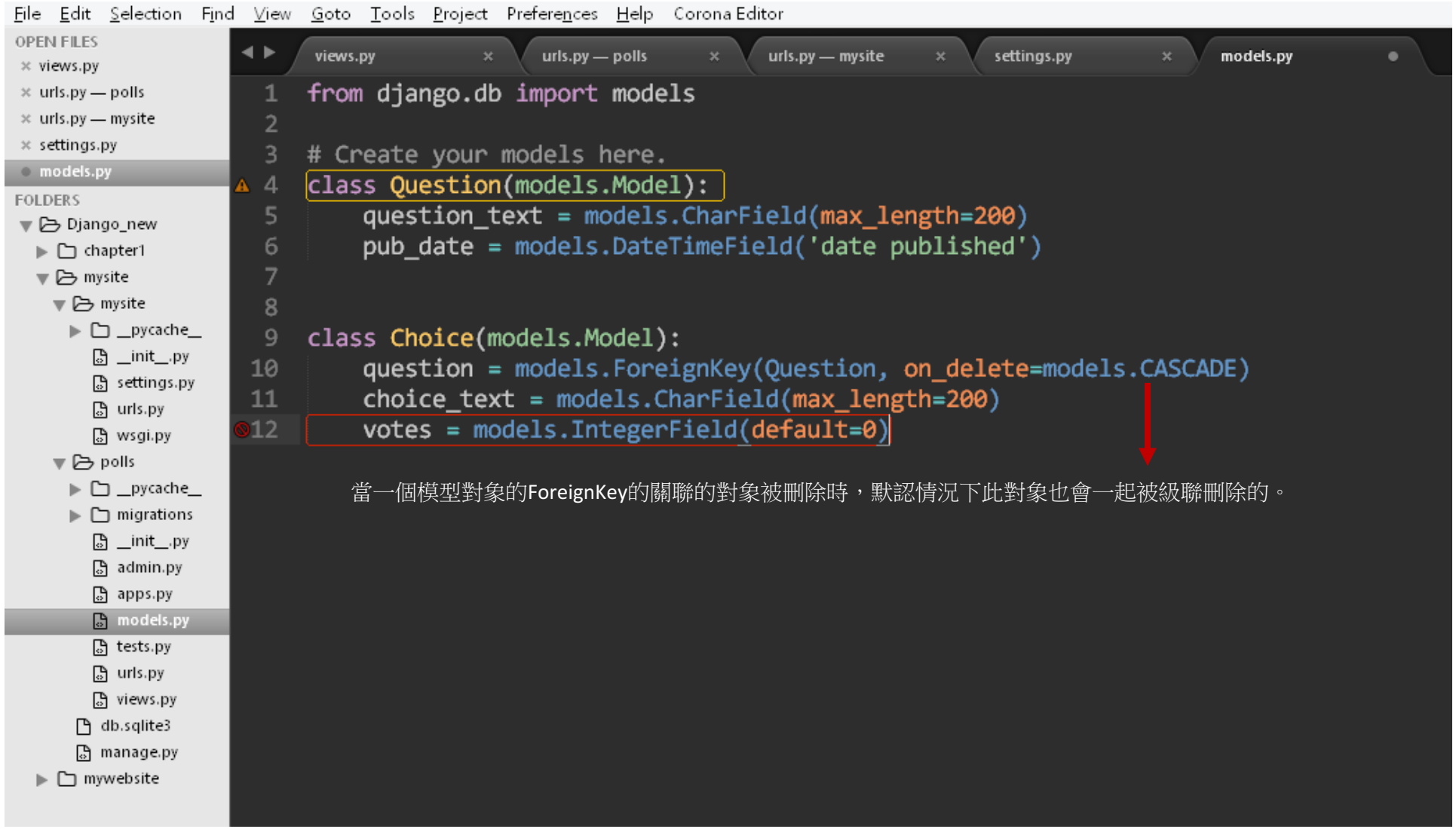
- The migrate command looks at the INSTALLED_APPS setting and creates any necessary database tables according to the database settings in your mysite/settings.py file and the database migrations shipped with the app.

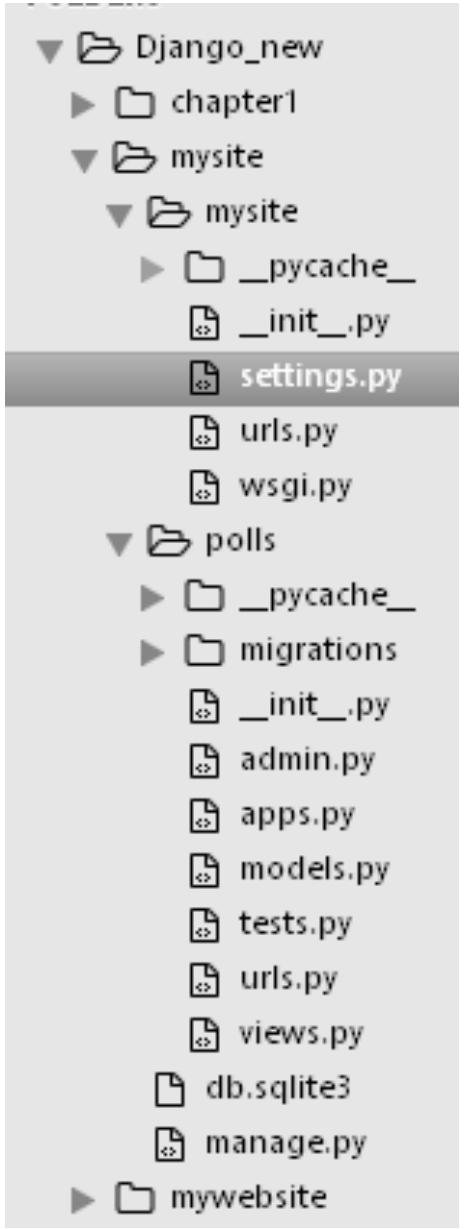


Other features:

- Data validation on data Saving/Update
- Form generation with help of form
- Auto generated Admin Site for adding, deleting, editing records

dsjerry2017@gmail.com


File Edit Selection Find View Goto Tools Project Preferences Help Corona Editor
OPEN FILES
views.py
urls.py — polls
urls.py — mysite
settings.py
models.py
FOLDERS
Django_new
chapter1
mysite
mysite
__pycache__
__init__.py
settings.py
urls.py
wsgi.py
polls
__pycache__
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
db.sqlite3
manage.py
mywebsite
views.py
urls.py — polls
urls.py — mysite
settings.py
models.py
1 from django.db import models
2
3 # Create your models here.
4 class Question(models.Model):
5 question_text = models.CharField(max_length=200)
6 pub_date = models.DateTimeField('date published')
7
8
9 class Choice(models.Model):
10 question = models.ForeignKey(Question, on_delete=models.CASCADE)
11 choice_text = models.CharField(max_length=200)
12 votes = models.IntegerField(default=0)
當一個模型對象的ForeignKey的關聯的對象被刪除時，默認情況下此對象也會一起被級聯刪除的。




```python
INSTALLED_APPS = [
    'polls.apps.PollsConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

# Internationalization
# https://docs.djangoproject.com/en/dev/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Asia/Taipei'

USE_I18N = True

USE_L10N = True

USE_TZ = True
```

- python manage.py makemigrations polls
- python manage.py sqlmigrate polls 0001
- python manage.py migrate

```
--
-- Create model Choice
--
CREATE TABLE "polls_choice" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "c
hoice_text" varchar(200) NOT NULL, "votes" integer NOT NULL);
--
-- Create model Question
--
CREATE TABLE "polls_question" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
"question_text" varchar(200) NOT NULL, "pub_date" datetime NOT NULL);
--
-- Add field question to choice
--
ALTER TABLE "polls_choice" RENAME TO "polls_choice__old";
CREATE TABLE "polls_choice" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "c
hoice_text" varchar(200) NOT NULL, "votes" integer NOT NULL, "question_id" integ
er NOT NULL REFERENCES "polls_question" ("id"));
INSERT INTO "polls_choice" ("question_id", "id", "choice_text", "votes") SELECT
NULL, "id", "choice_text", "votes" FROM "polls_choice__old";
DROP TABLE "polls_choice__old";
CREATE INDEX "polls_choice_question_id_c5b4b260" ON "polls_choice" ("question_id
");
COMMIT;
```

Playing with the API

- python manage.py shell
- **from** polls.models **import** Question, Choice
- Question.objects.all()
- **from** django.utils **import** timezone
- q **=** Question(question_text**=**“What‘s new", pub_date**=**timezone.now())
- q.save()
- q.id
- q.question_text
- q.pub_date

Playing with the API

- q.question_text = "What's up?“
- q.save()
- Question.objects.all()

```
>>> Question.objects.all()
<QuerySet [<Question: Question object>]>
>>>
```

Playing with the API

```python
from django.db import models

# Create your models here.
class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

    def __str__(self):
        return self.question_text


class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)

    def __str__(self):
        return self.choice_text
```

Playing with the API

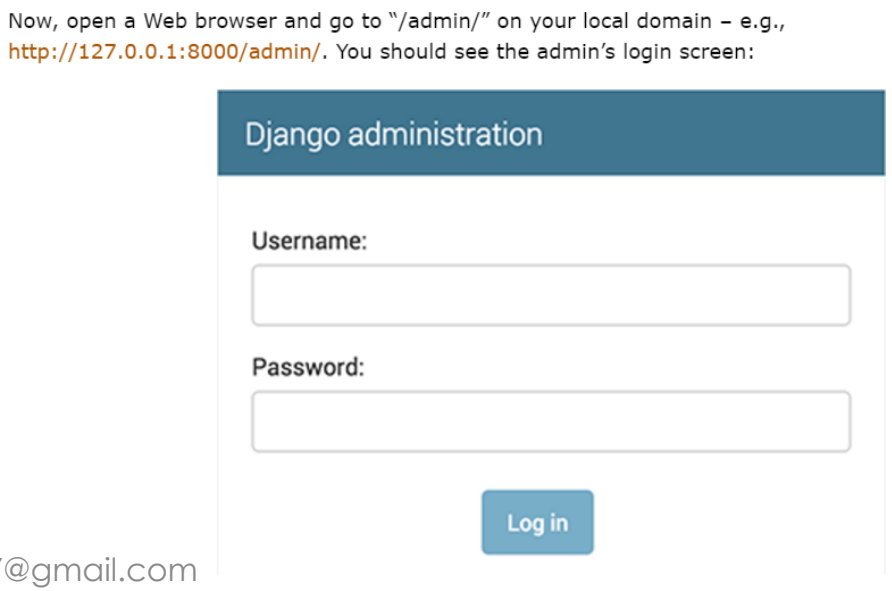
- python manage.py makemigrations polls
- python manage.py migrate
- python manage.py shell

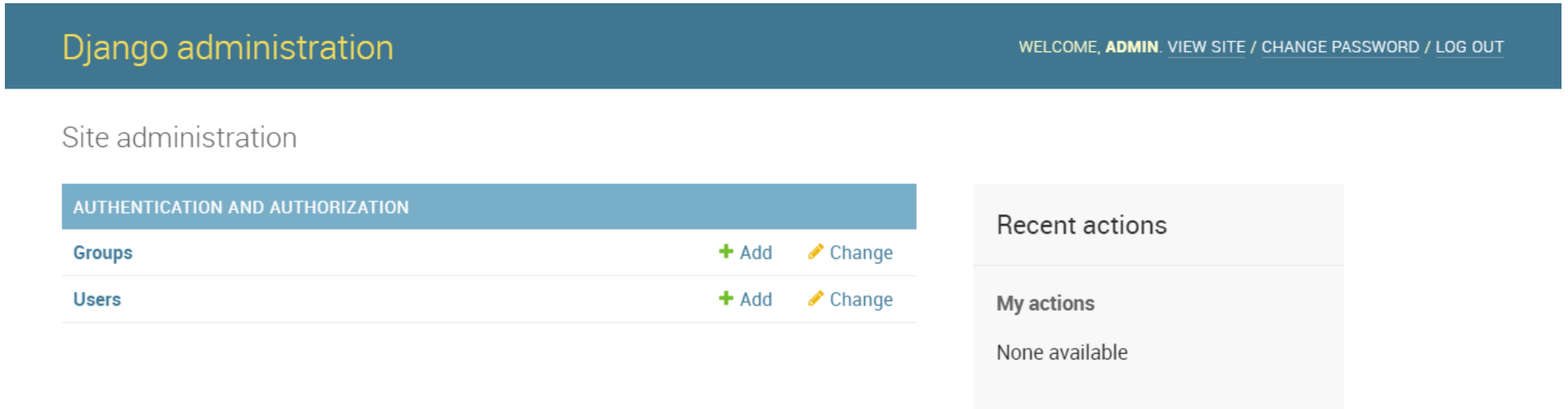
- **from** polls.models **import** Question, Choice
- Question.objects.all()

```
>>> from polls.models import Question, Choice
>>> Question.objects.all()
<QuerySet [<Question: What's up>]>
```

Creating an admin user

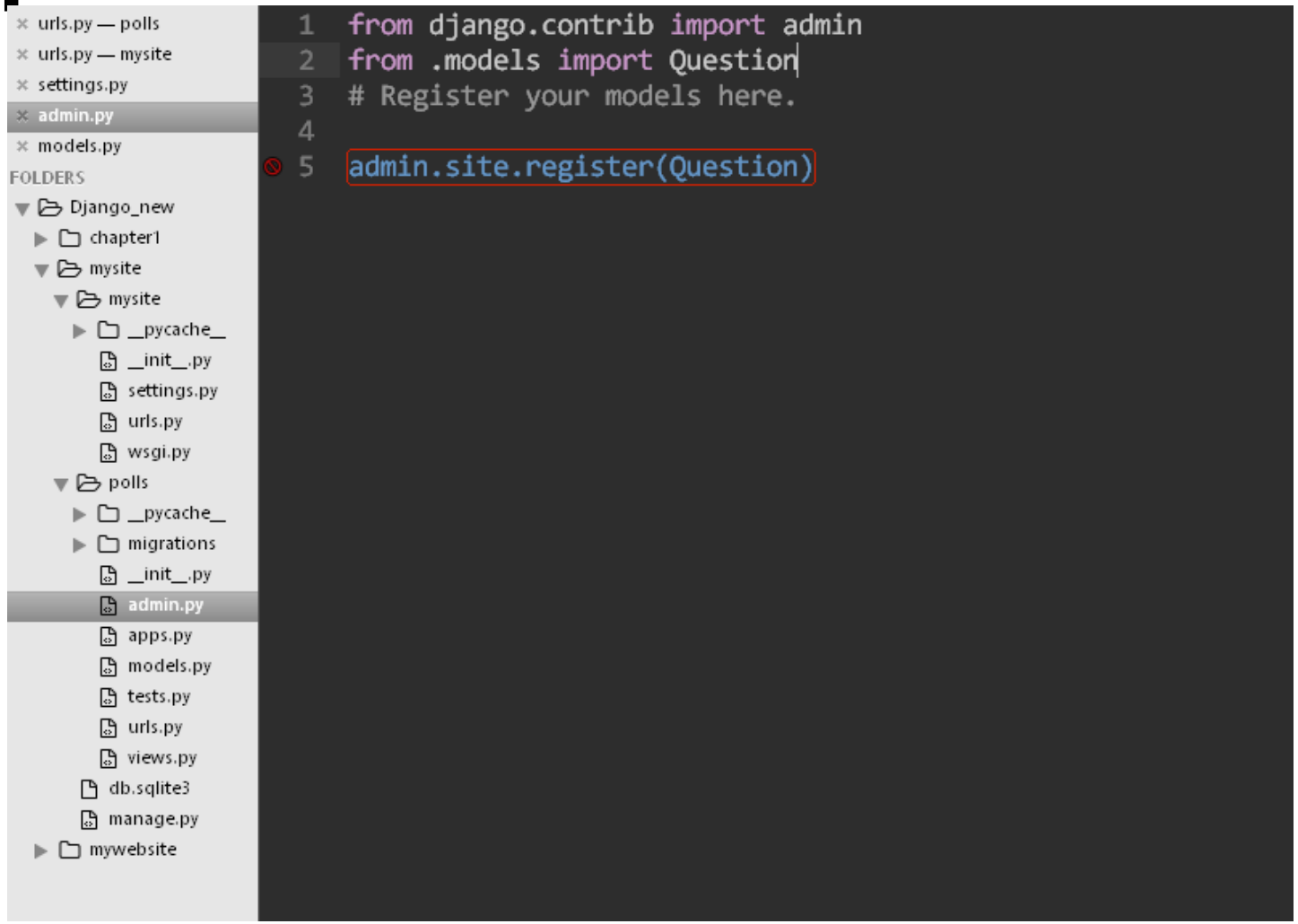
- python manage.py createsuperuser
- Username: admin
- Email address: admin@example.com
- Password: &1234567
- python manage.py runserve

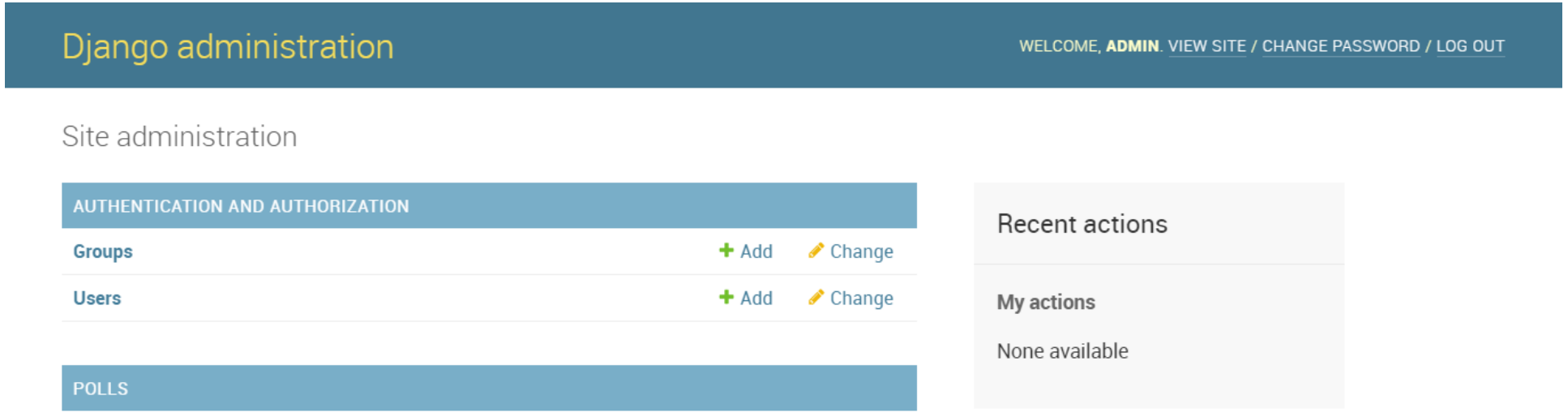




dsjerry2017@gmail.com

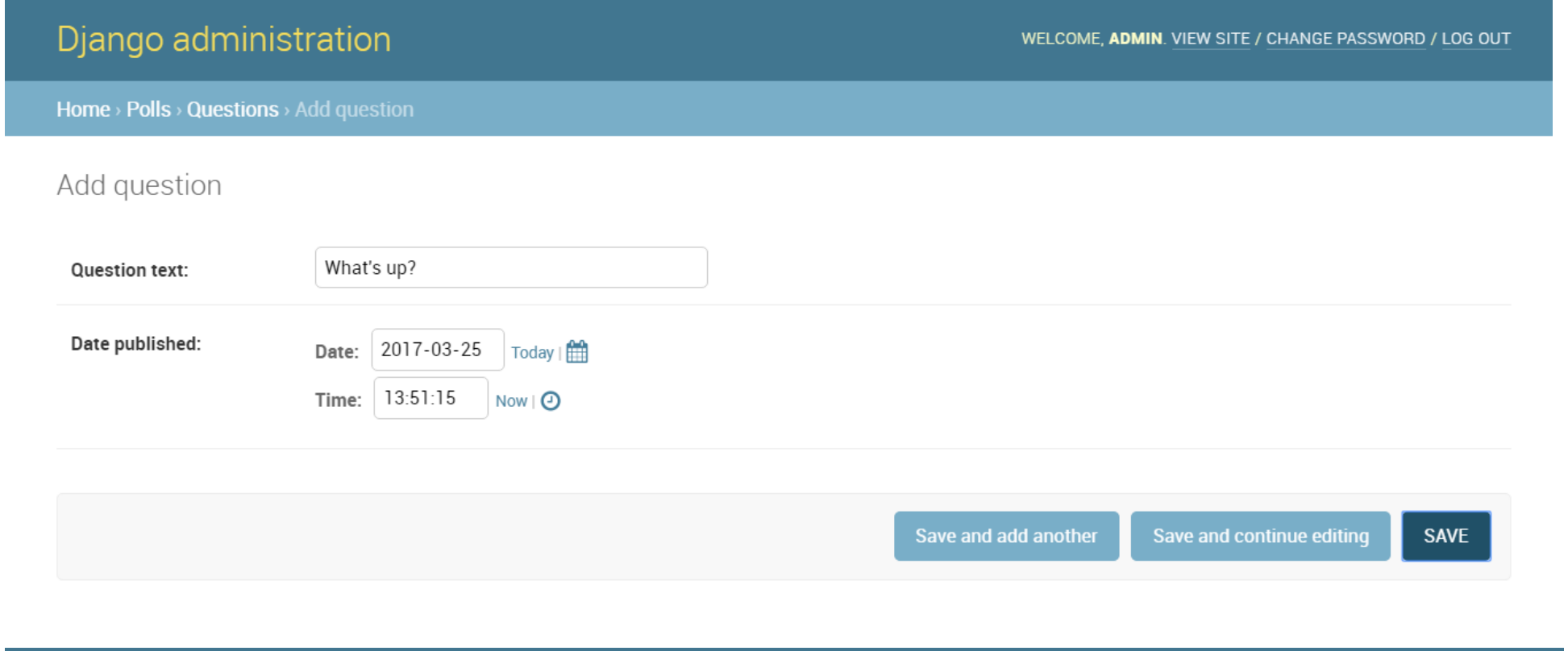
Make the poll app modifiable in the admin

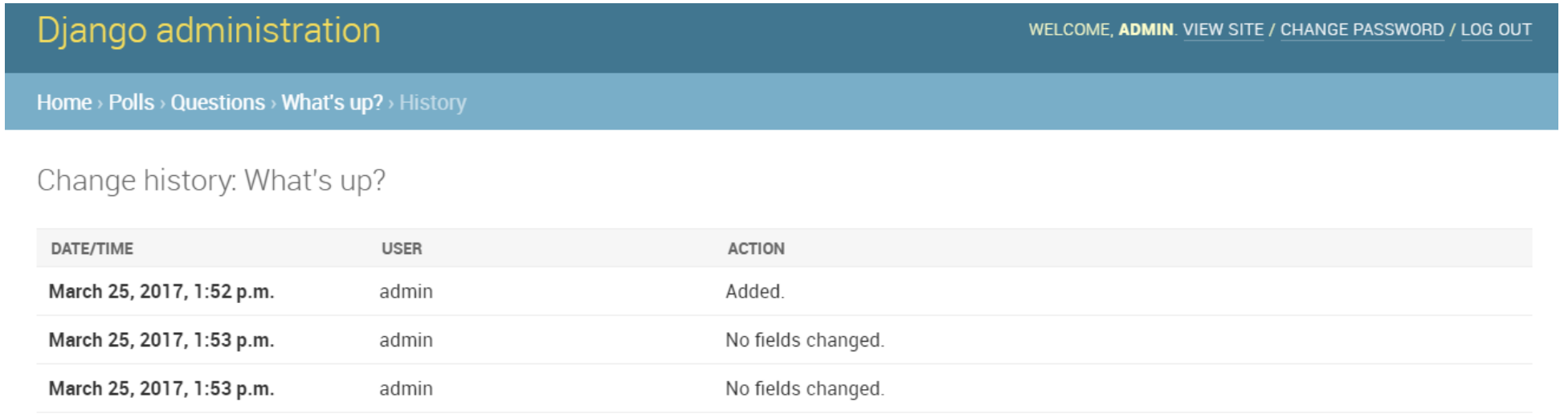




dsjerry2017@gmail.com

Z





dsjerry2017@gmail.com

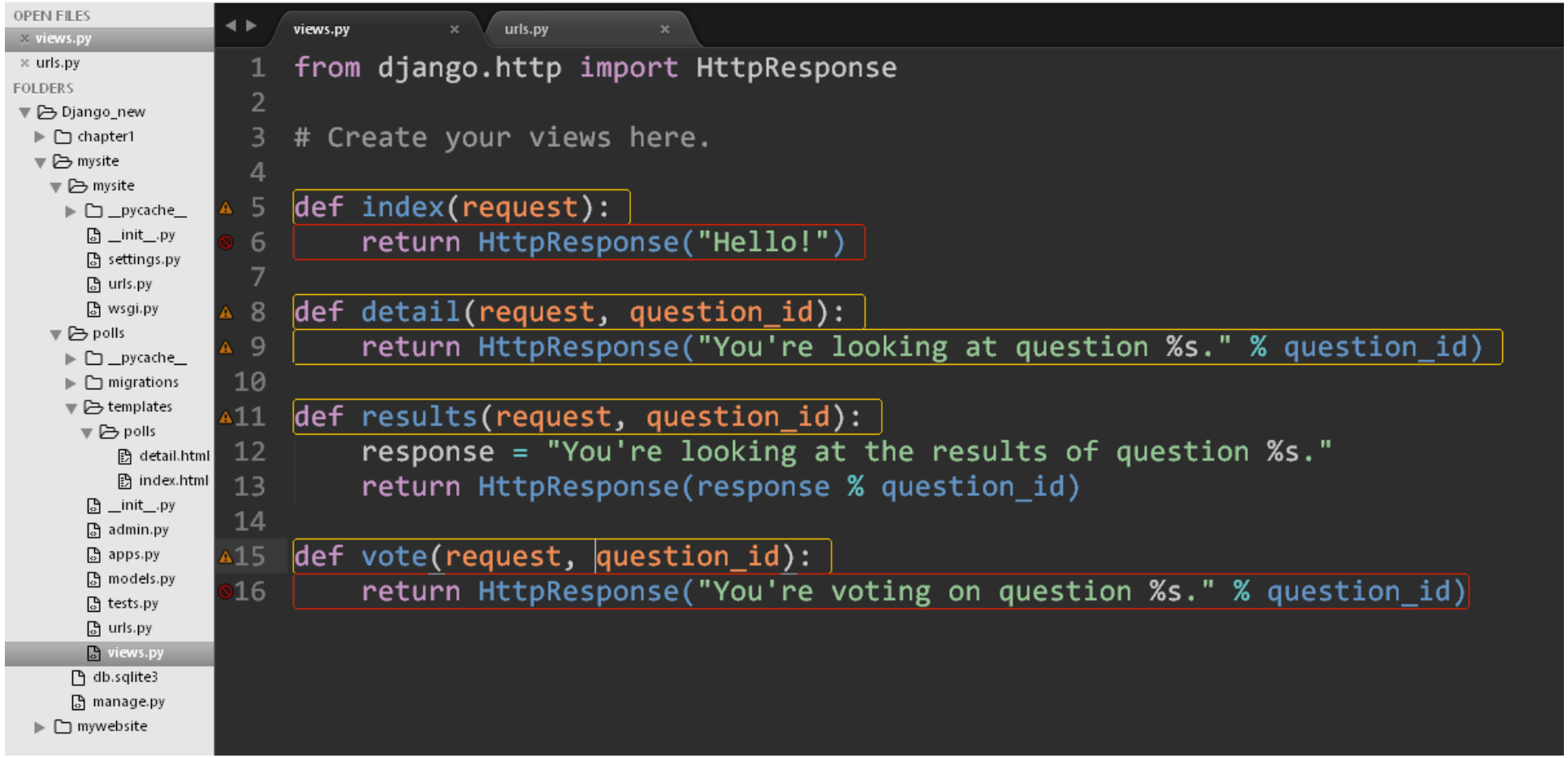
Part 3: Views and templates

Functions

- Question “index” page – displays the latest few questions.
- Question “detail” page – displays a question text, with no results but with a form to vote.
- Question “results” page – displays results for a particular question.
- Vote action – handles voting for a particular choice in a particular question.

polls/urls.py

polls/views.py



常見錯誤

```
 6  # Create your views here.
 7  def index(request):
 8      return HttpResponse("Hello, Polls. 你好 投票系統")
 9
10          def detail(request, question_id):
11              return HttpResponse("You're looking at ques
```

1. python3 manage.py runserver (Python)

```
kages/django/conf/urls/__init__.py", line 50, in include
    urlconf_module = import_module(urlconf_module)
  File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/importli
b/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 978, in _gcd_import
  File "<frozen importlib._bootstrap>", line 961, in _find_and_load
  File "<frozen importlib._bootstrap>", line 950, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 655, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 678, in exec_module
  File "<frozen importlib._bootstrap>", line 205, in _call_with_frames_removed
  File "/Users/jerry/Documents/git/class2/mysite/polls/urls.py", line 3, in <mod
ule>
    from . import views
  File "/Users/jerry/Documents/git/class2/mysite/polls/views.py", line 10
    def detail(request, question_id):
                                    ^
TabError: inconsistent use of tabs and spaces in indentation
```

常見錯誤

```
 6  # Create your views here.
 7  def index(request):
 8      return HttpResponse("Hello, Polls. 你好 投票系統")
 9
10  def detail(request, question_id):
11  return HttpResponse("You're looking at question %s." %
```

1. python3 manage.py runserver (Python)

```
kages/django/conf/urls/__init__.py", line 50, in include
    urlconf_module = import_module(urlconf_module)
  File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/importli
b/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 978, in _gcd_import
  File "<frozen importlib._bootstrap>", line 961, in _find_and_load
  File "<frozen importlib._bootstrap>", line 950, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 655, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 678, in exec_module
  File "<frozen importlib._bootstrap>", line 205, in _call_with_frames_removed
  File "/Users/jerry/Documents/git/class2/mysite/polls/urls.py", line 3, in <mod
ule>
    from . import views
  File "/Users/jerry/Documents/git/class2/mysite/polls/views.py", line 11
    return HttpResponse("You're looking at question %s." % question_id)
         ^
IndentationError: expected an indented block
```

常見錯誤

```
 6  # Create your views here.
 7  def index(request):
 8      return HttpResponse("Hello, Polls. 你好 投票系統")
 9
10  def detail(request, question_id):
11              return HttpResponse("You're looking at ques
```

1. python3 manage.py runserver (Python)

```
odule>
    url(r'^', include('polls.urls')),
  File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-pac
kages/django/conf/urls/__init__.py", line 50, in include
    urlconf_module = import_module(urlconf_module)
  File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/importli
b/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 978, in _gcd_import
  File "<frozen importlib._bootstrap>", line 961, in _find_and_load
  File "<frozen importlib._bootstrap>", line 950, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 655, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 678, in exec_module
  File "<frozen importlib._bootstrap>", line 205, in _call_with_frames_removed
  File "/Users/jerry/Documents/git/class2/mysite/polls/urls.py", line 8, in <mod
ule>
    url(r'^(?P<question_id>[0-9]+)/results/$', views.results, name='results'),
AttributeError: module 'polls.views' has no attribute 'results'
```

```python
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^(?P<question_id>[0-9]+)/$', views.detail, name='detail'),
    url(r'^(?P<question_id>[0-9]+)/results/$', views.results, name='results'),
    url(r'^(?P<question_id>[0-9]+)/vote/$', views.vote, name='vote'),
]
```

http://127.0.0.1:8000/polls/22/

```python
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^(?P<question_id>[0-9]+)/$', views.detail, name='detail'),
    url(r'^(?P<question_id>[0-9]+)/results/$', views.results, name='results'),
    url(r'^(?P<question_id>[0-9]+)/vote/$', views.vote, name='vote'),
]
```

http://127.0.0.1:8000/polls/1/results/

You're looking at the results of question 1.

```python
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^(?P<question_id>[0-9]+)/$', views.detail, name='detail'),
    url(r'^(?P<question_id>[0-9]+)/results/$', views.results, name='results'),
    url(r'^(?P<question_id>[0-9]+)/vote/$', views.vote, name='vote'),
]
```

http://127.0.0.1:8000/polls/1/results/

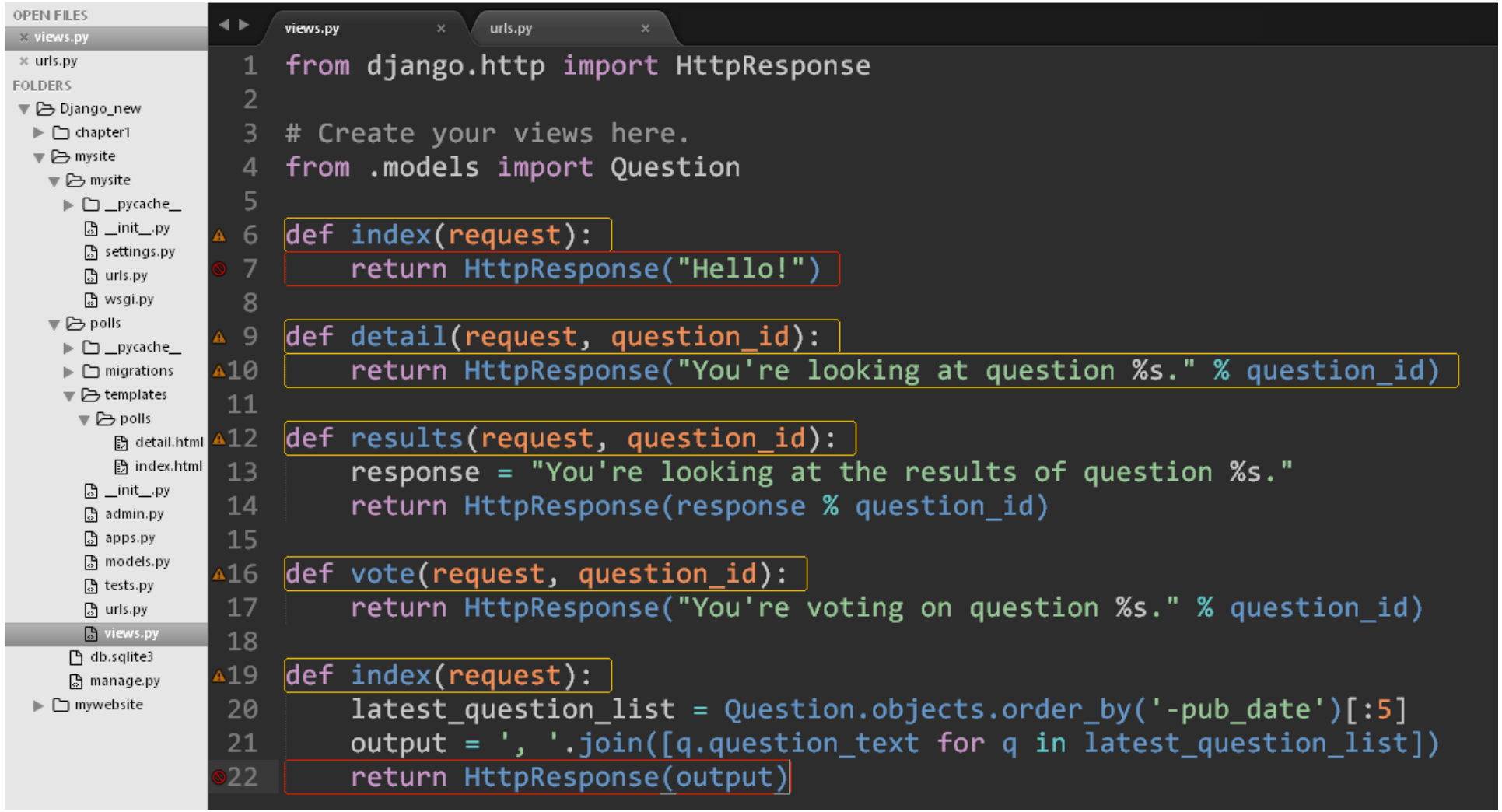
127.0.0.1:8000/polls/1/vote/

You're voting on question 1.

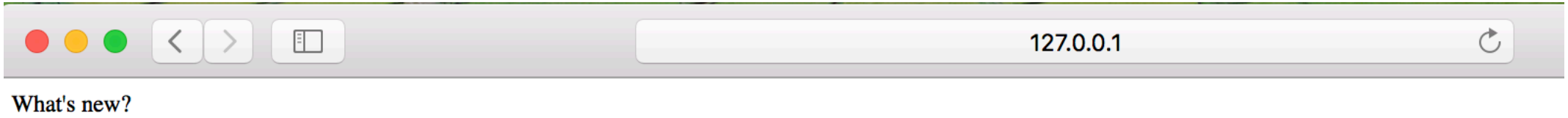
Write views that actually do something

- Each view is responsible for doing one of two things: returning an HttpResponse object containing the content for the requested page, or raising an exception such as Http404.

polls/views.py



http://127.0.0.1:8000/jerry/5/vote/



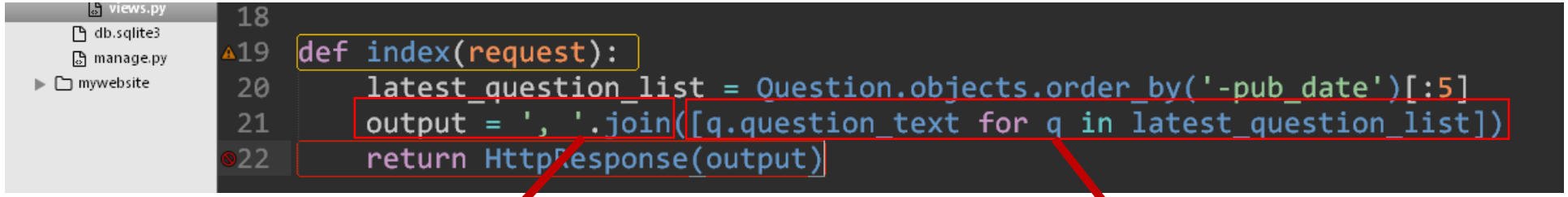
Python Shell

views.py
db.sqlite3
manage.py
mywebsite

```
18
19  def index(request):
20      latest_question_list = Question.objects.order_by('-pub_date')[:5]
21      output = ', '.join([q.question_text for q in latest_question_list])
22      return HttpResponse(output)
```

```
In [1]: from polls.models import Question, Choice

In [2]: Question.objects.all()
Out[2]: <QuerySet [<Question: 這是問題1>, <Question: 問題2>]>

In [3]: Question.objects.order_by('-pub_date')[:5]
Out[3]: <QuerySet [<Question: 問題2>, <Question: 這是問題1>]>

In [4]: Question.objects.order_by('pub_date')[:5]
Out[4]: <QuerySet [<Question: 這是問題1>, <Question: 問題2>]>

In [5]:
```

Python Shell

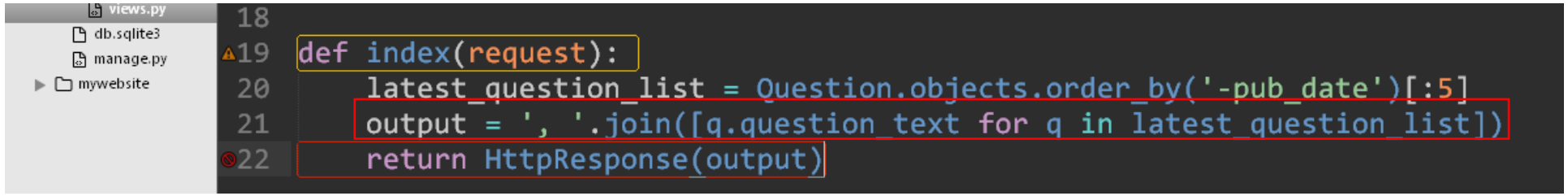



```
In [30]: x = ['my','name','is','bob']

In [31]: ', '.join(x)
Out[31]: 'my, name, is, bob'

In [32]:
```

```
In [41]: for i in x:
    ...:     print(i)
    ...:
my
name
is
bob
```

```
In [44]: x_list = [i for i in x]

In [45]: print(x_list)
['my', 'name', 'is', 'bob']
```

Python Shell




```
In [46]: ', '.join( [i for i in x] )
Out[46]: 'my, name, is, bob'

In [47]: join_list = ', '.join( [i for i in x] )

In [48]: type(join_list)
Out[48]: str

In [49]:
```

Python Shell

views.py
db.sqlite3
manage.py
mywebsite

```
18
19  def index(request):
20      latest_question_list = Question.objects.order_by('-pub_date')[:5]
21      output = ', '.join([q.question_text for q in latest_question_list])
22      return HttpResponse(output)
```

```
In [63]: latest_question_list = Question.objects.order_by('-pub_date')[:5]

In [64]: ', '.join([q.question_text for q in latest_question_list])
Out[64]: '問題2, 這是問題1'

In [65]: [q.question_text for q in latest_question_list]
Out[65]: ['問題2', '這是問題1']

In [66]: output = ', '.join([q.question_text for q in latest_question_list])

In [67]: HttpResponse(output)
Out[67]: <HttpResponse status_code=200, "text/html; charset=utf-8">

In [68]: print(output)
問題2, 這是問題1

In [69]:
```

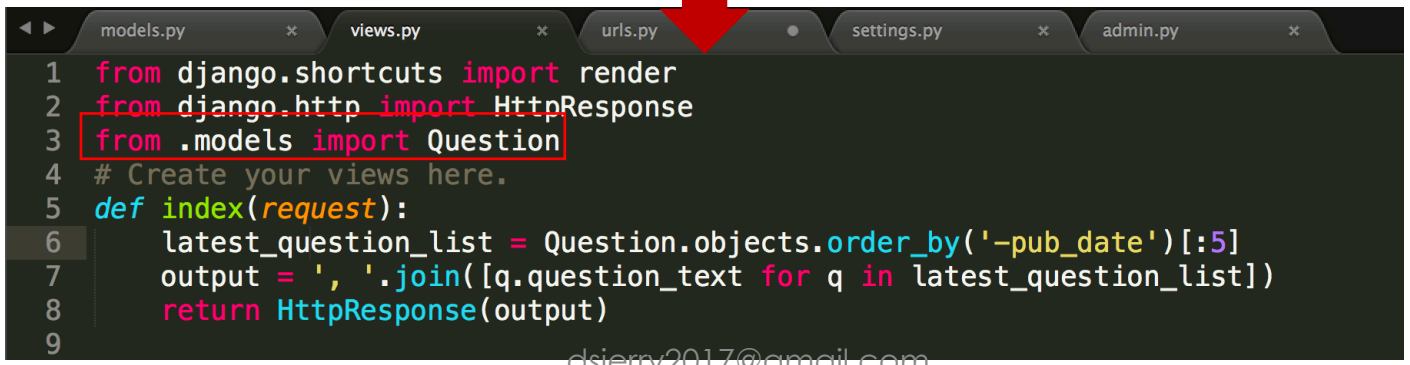
把問題標題，轉成字串輸出。

常見錯誤

NameError at /polls/

name 'Question' is not defined

Request Method:	GET
Request URL:	http://127.0.0.1:8000/polls/
Django Version:	1.11rc1
Exception Type:	NameError
Exception Value:	`name 'Question' is not defined`
Exception Location:	/Users/jerry/Documents/git/class2/Chapter1_test/polls/views.py in index, line 6
Python Executable:	/usr/local/bin/python3
Python Version:	3.6.0
Python Path:	`['/Users/jerry/Documents/git/class2/Chapter1_test', '/Library/Frameworks/Python.framework/Versions/3.6/lib/python36.zip', '/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6', '/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/lib-dynload', '/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages']`
Server time:	Wed, 10 May 2017 23:48:46 +000



Home › Polls › Questions

The question "五月天厲害嗎?" was added successfully.

Select question to change

ADD QUESTION +

Action: --------- Go 0 of 3 selected

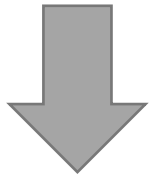
	QUESTION
☐	五月天厲害嗎?
☐	What's up?
☐	What's up

3 questions

如果要編輯頁面的內容，要如何做？
第一步驟 加上Templates
第二步驟 加上Render

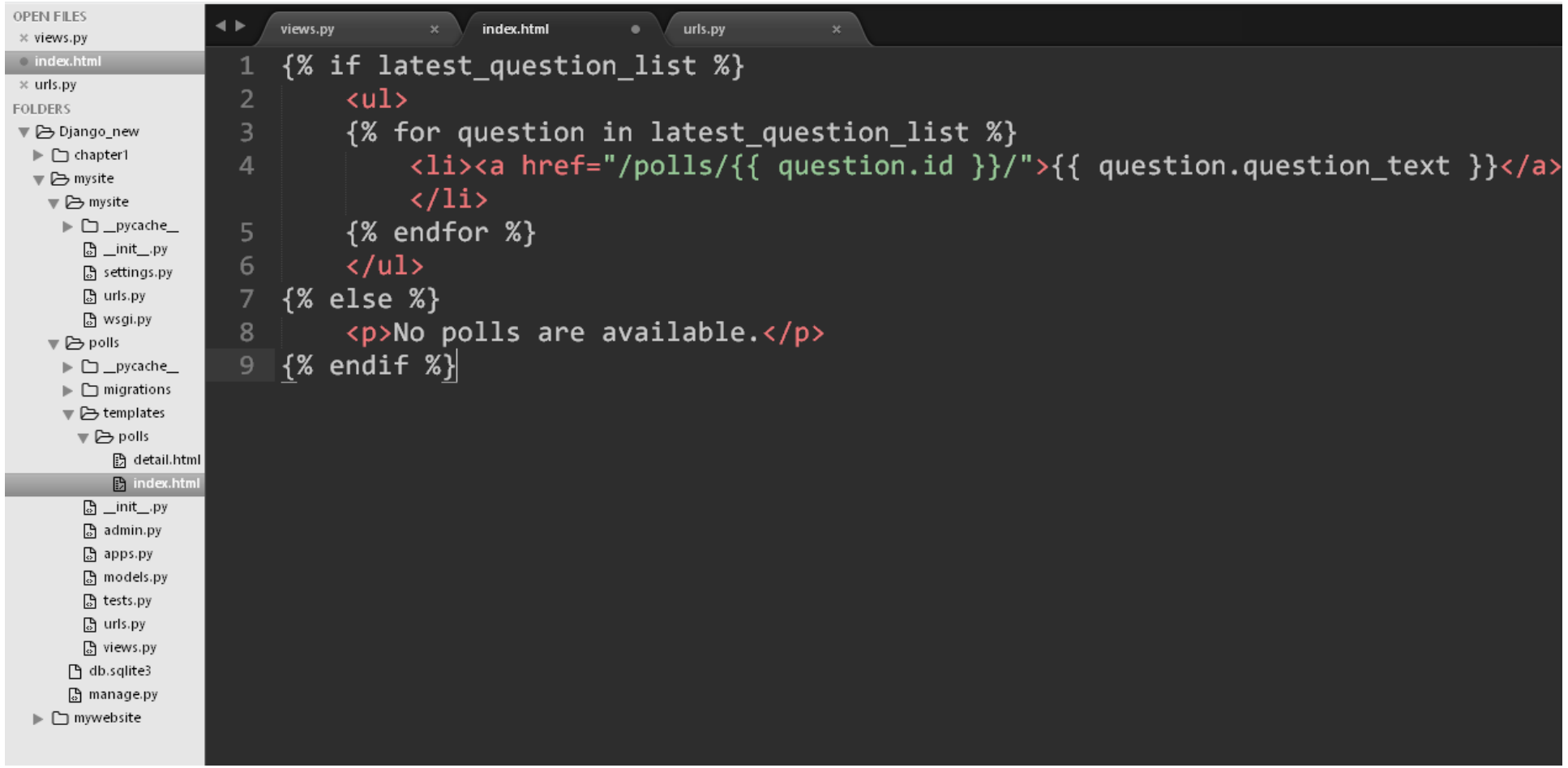
dsjerry2017@gmail.com


127.0.0.1:8000/polls/
應用程式
多个地点Ping服务器
Quarca - Responsive
五月天厲害嗎?, What's up?, What's up



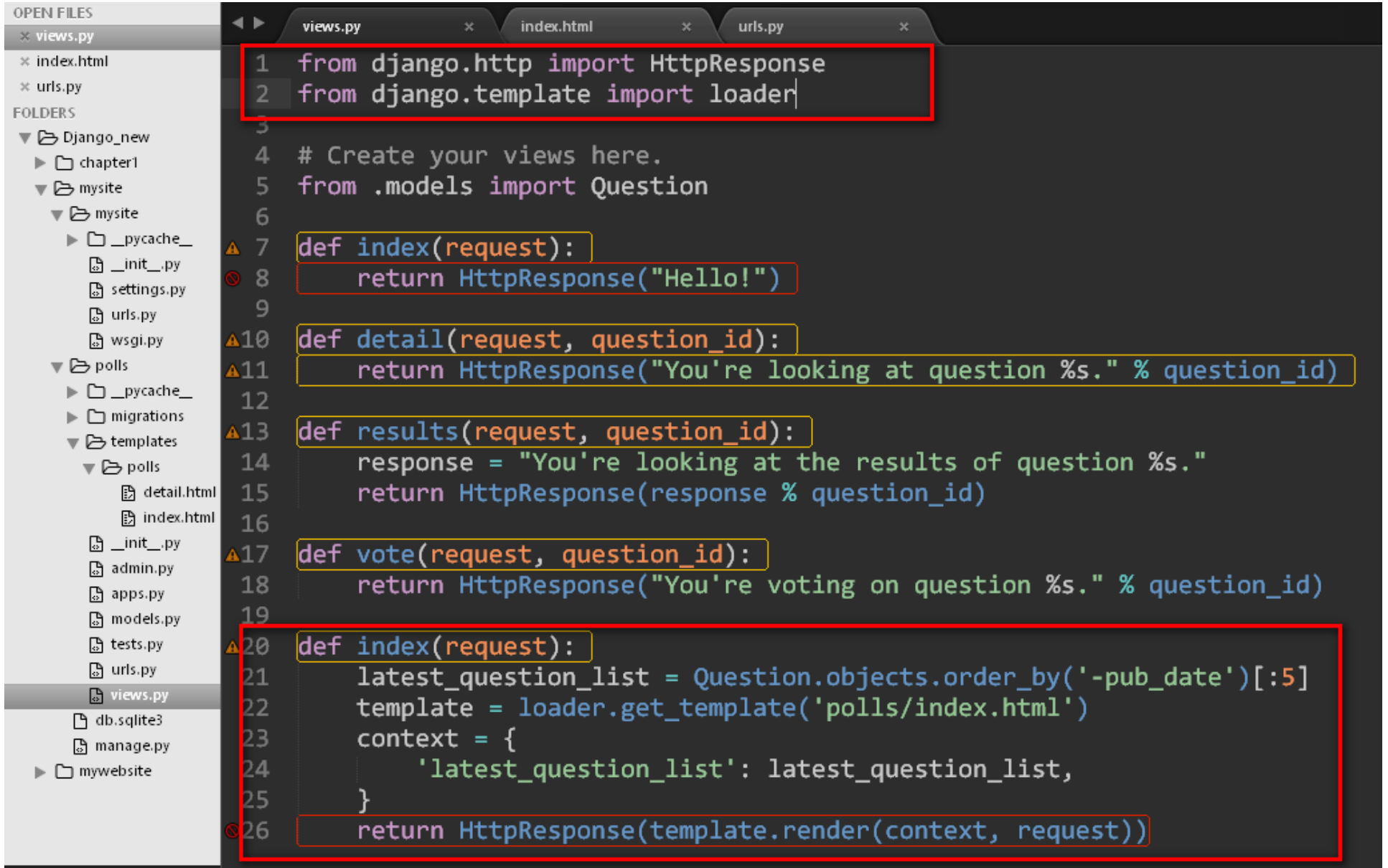

應用程式
多个地点Ping服务器
Quarca - Responsive
投票網站
五月天厲害嗎?
What's up?
What's up

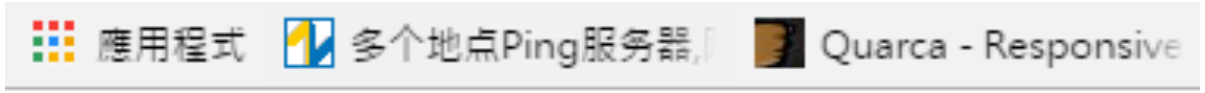
polls/templates/polls/index.html



polls/views.py

dsjerry2017@gmail.com





投票網站

- 五月天厲害嗎?
- What's up?
- What's up

views.py | index.html | urls.py

```
<h1>投票網站</h1>
{% if latest_question_list %}
    <ul>
    {% for question in latest_question_list %}
        <li><a href="/polls/{{ question.id }}/">{{ question.
        question_text }}</a></li>
    {% endfor %}
    </ul>
{% else %}
    <p>No polls are available.</p>
{% endif %}
```

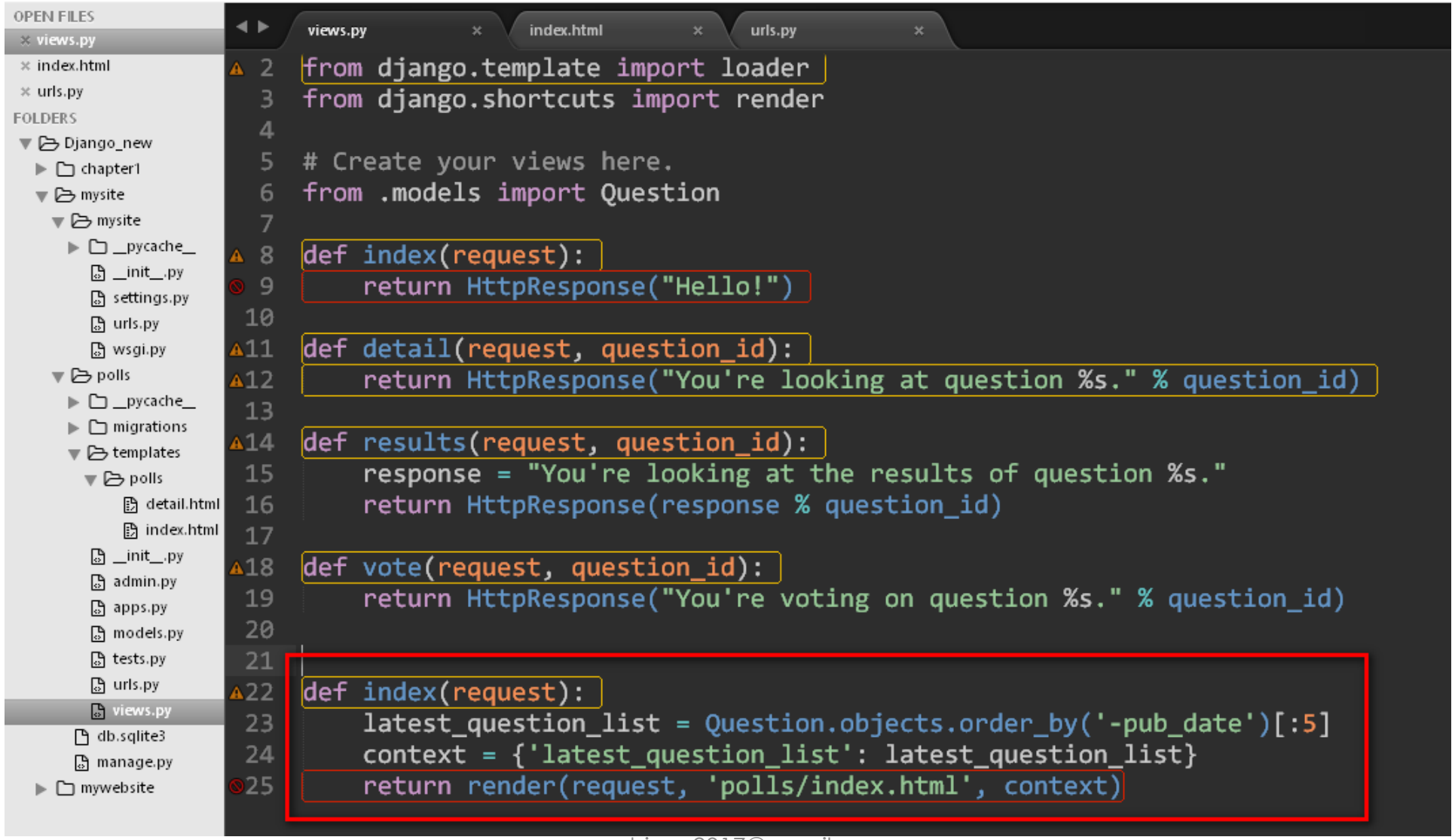
A shortcut: render()

```python
21  def index(request):
22      latest_question_list = Question.objects.order_by('-pub_date')[:5]
23      template = loader.get_template('polls/index.html')
24      context = {
25          'latest_question_list': latest_question_list,
26      }
27      return HttpResponse(template.render(context, request))
```

設定Templates的位置

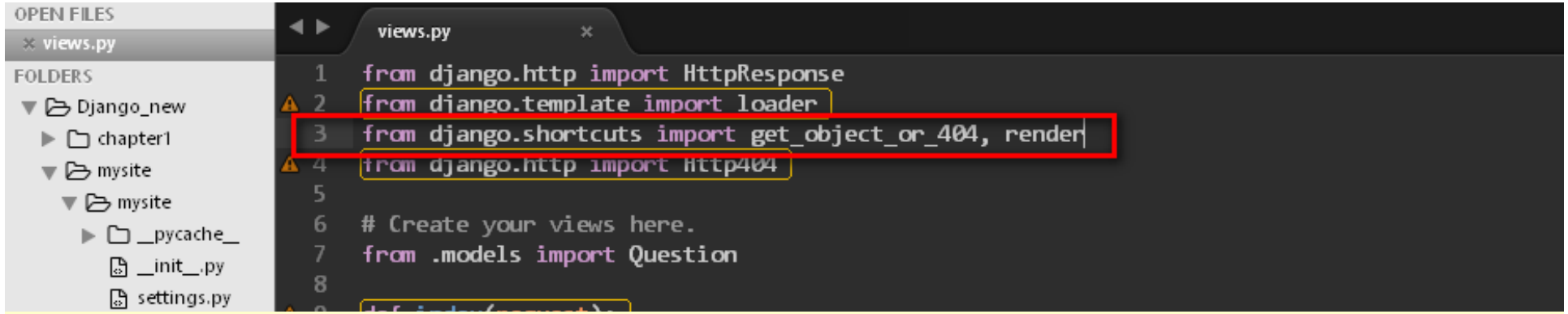
```python
29  def index(request):
30      latest_question_list = Question.objects.order_by('-pub_date')[:5]
31      context = {'latest_question_list': latest_question_list}
32      return render(request, 'polls/index.html', context)
```

A shortcut: render()



A shortcut: get_object_or_404()

dsjerry2017@gmail.com

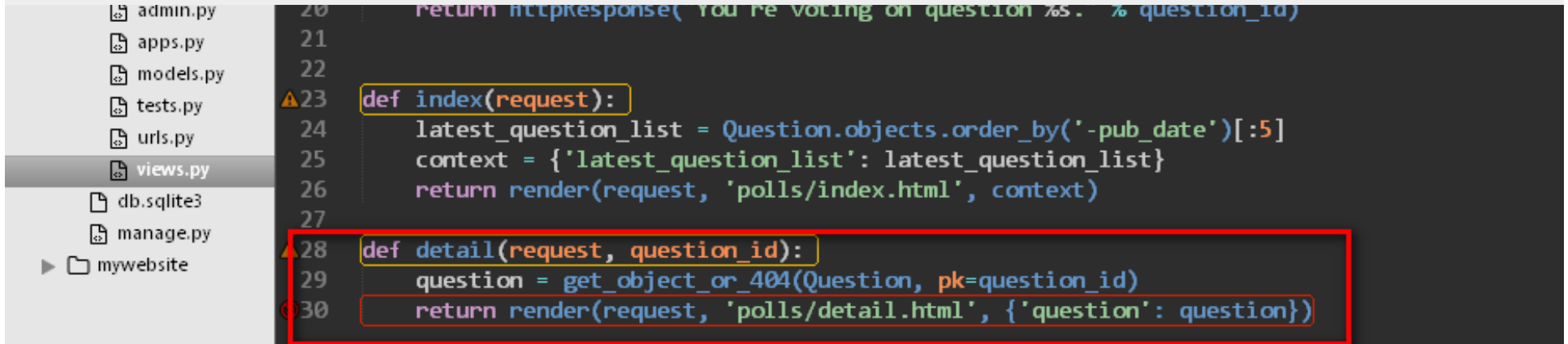


Page not found (404)

Request Method: GET
Request URL: http://127.0.0.1:8000/polls/11/
Raised by: polls.views.detail

No Question matches the given query.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



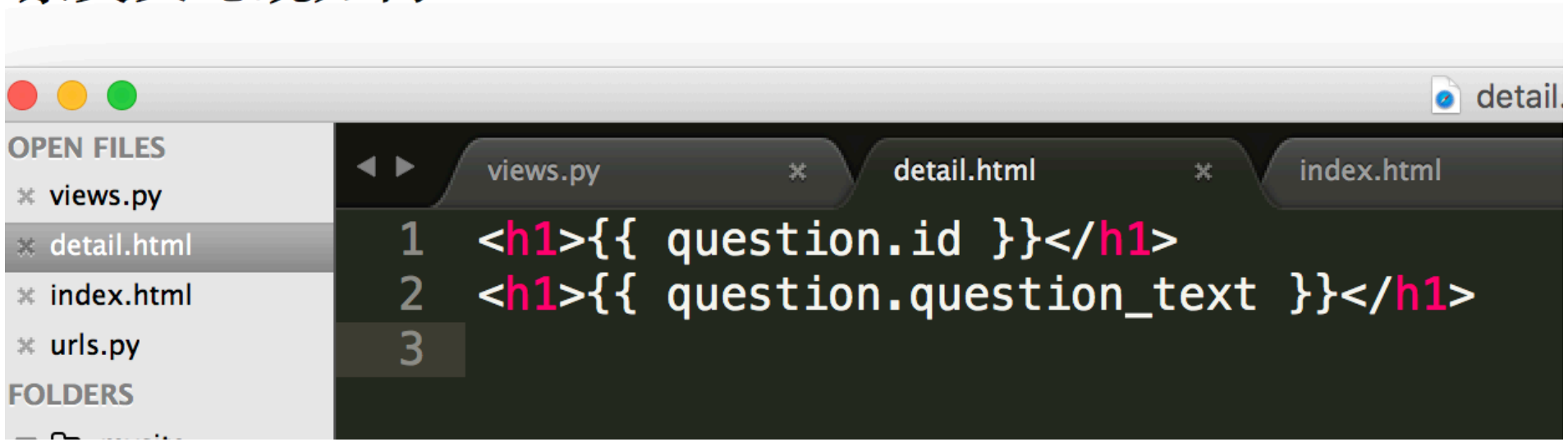
務必記得加上detail.html

polls/templates/polls/detail.html

```
{{ question }}
```

3

蔡英文總統如何？


detail.
OPEN FILES
views.py
detail.html
index.html
urls.py
FOLDERS
views.py
detail.html
index.html
1 <h1>{{ question.id }}</h1>
2 <h1>{{ question.question_text }}</h1>
3

Namespacing URL names

dsjerry2017@gmail.com

- How does one make it so that Django knows which app view to create for a url when using the {% url %} template tag？ (方便識別自己用的APP)

polls/urls.py

```
from django.conf.urls import url

from . import views

app_name = 'polls'
urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^(?P<question_id>[0-9]+)/$', views.detail, name='detail'),
    url(r'^(?P<question_id>[0-9]+)/results/$', views.results, name='results'),
    url(r'^(?P<question_id>[0-9]+)/vote/$', views.vote, name='vote'),
]
```

polls/templates/polls/index.html

```
<li><a href="{% url 'polls:detail' question.id %}">{{ question.question_text }}</a></li>
```

Part 4: Forms and generic views

dsjerry2017@gmail.com

HTML Forms

- **action** :The **action** attribute defines the action to be performed when the form is submitted.
  Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

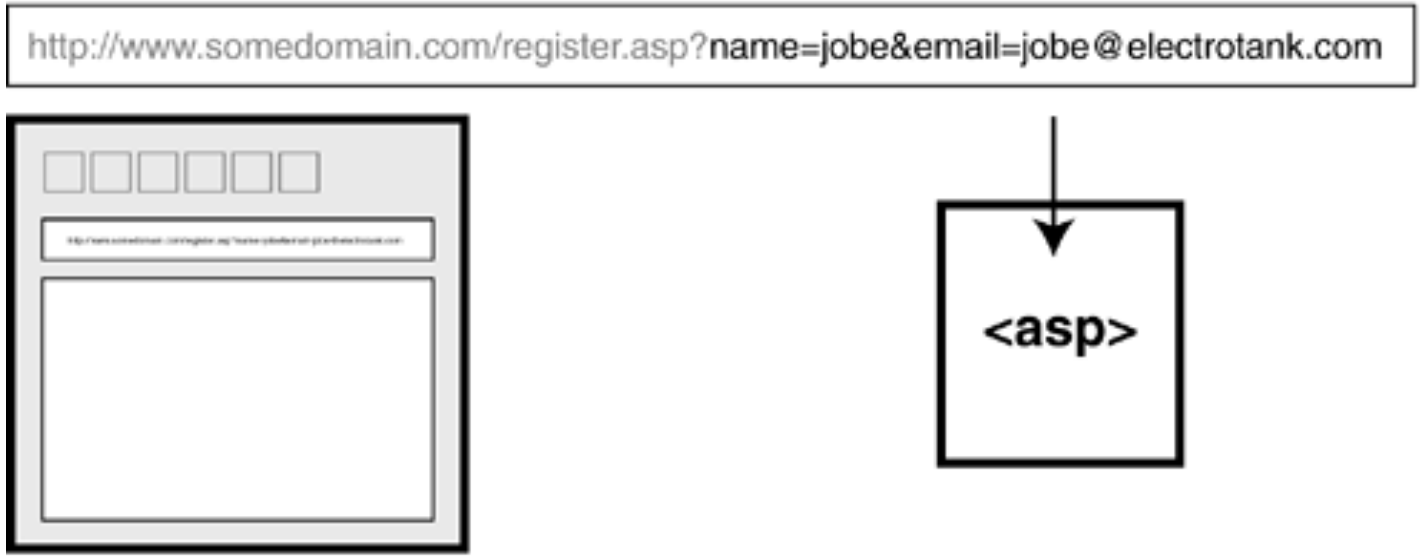
```
1  <h1>{{ question.question_text }}</h1>
2
3  {% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}
4
5  <form action="{% url 'polls:vote' question.id %}" method="post">
6  {% csrf_token %}
7  {% for choice in question.choice_set.all %}
8      <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />
9      <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br />
10 {% endfor %}
11 <input type="submit" value="Vote" />
12 </form>
```

HTML Forms

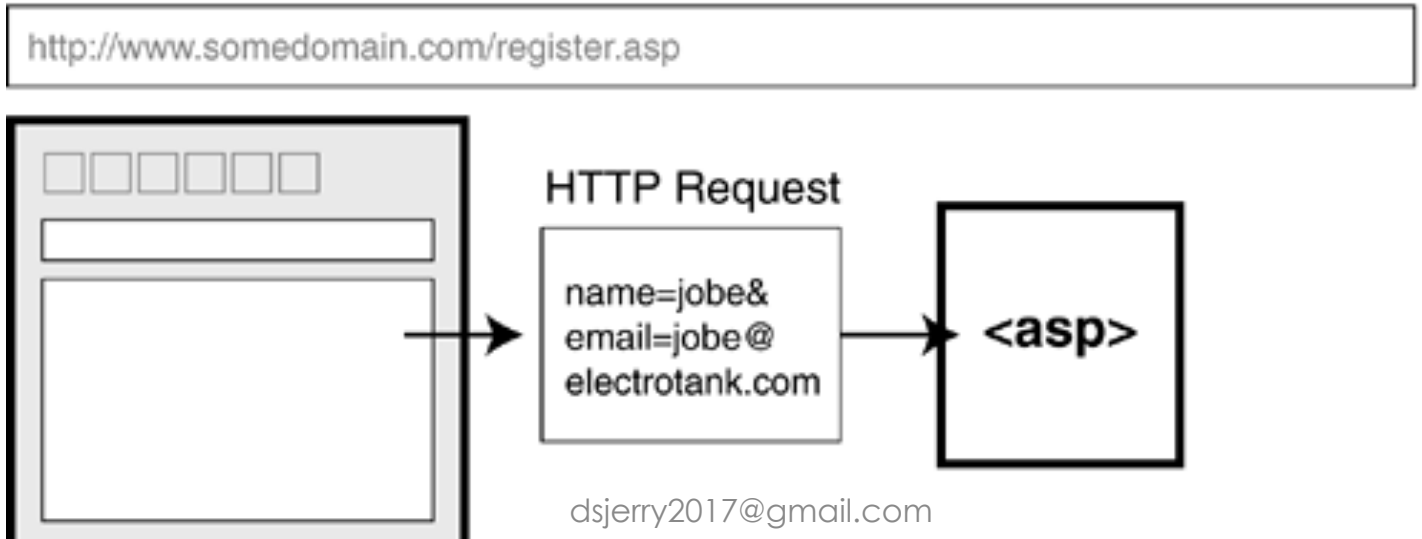
- The **method** attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data

```
1  <h1>{{ question.question_text }}</h1>
2
3  {% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}
4
5  <form action="{% url 'polls:vote' question.id %}" method="post">
6  {% csrf_token %}
7  {% for choice in question.choice_set.all %}
8      <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />
9      <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br />
10 {% endfor %}
11 <input type="submit" value="Vote" />
12 </form>
```

Using GET



Using POST



HTML Forms (detail.html)

- **<input type="radio">** defines a **radio button**.
- Radio buttons let a user select ONLY ONE of a limited number of choices:

```
1  <h1>{{ question.question_text }}</h1>
2
3  {% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}
4
5  <form action="{% url 'polls:vote' question.id %}" method="post">
6  {% csrf_token %}
7  {% for choice in question.choice_set.all %}
8      <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />
9      <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br />
10 {% endfor %}
11 <input type="submit" value="Vote" />
12 </form>
```

```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  <input type="radio" name="gender" value="male"> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other<br><br>
  <input type="submit">
</form>
```

Male
Female
Other

提交

```python
from django.http import HttpResponseRedirect, HttpResponse
from django.template import loader
from django.shortcuts import get_object_or_404, render
from django.http import Http404
from django.urls import reverse
```

```python
def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        # Redisplay the question voting form.
        return render(request, 'polls/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        # Always return an HttpResponseRedirect after successfully dealing
        # with POST data. This prevents data from being posted twice if a
        # user hits the Back button.
        return HttpResponseRedirect(reverse('polls:results', args=(question.id,)))
```

Add Options

加入選項

```python
from django.contrib import admin

# Register your models here.
from .models import Choice, Question

class ChoiceInline(admin.TabularInline):
    model = Choice
    extra = 3

class QuestionAdmin(admin.ModelAdmin):
    fieldsets = [
        (None,               {'fields': ['question_text']}),
        ('Date information', {'fields': ['pub_date'], 'classes': ['collapse']}),
    ]
    inlines = [ChoiceInline]
    list_display = ('question_text', 'pub_date')
    list_filter = ['pub_date']
    search_fields = ['question_text']


admin.site.register(Question, QuestionAdmin)
```

Add Results

Results.html

views.py | detail.html | results.html

```
<h1>{{ question.question_text }}</h1>

<ul>
{% for choice in question.choice_set.all %}
    <li>{{ choice.choice_text }} -- {{ choice.votes }} vote{{ choice.votes|pluralize }}</li>
{% endfor %}
</ul>

<a href="{% url 'polls:detail' question.id %}">Vote again?</a>
```

Views.py Check

```python
from django.shortcuts import render
from django.http import HttpResponseRedirect, HttpResponse
from django.urls import reverse
from django.shortcuts import get_object_or_404, render
from .models import Choice, Question

# Create your views here.
def index(request):
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
    context = {'latest_question_list': latest_question_list}
    return render(request, 'jerry/index.html', context)

def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'jerry/detail.html', {'question': question})

def results(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'jerry/results.html', {'question': question})

def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        # Redisplay the question voting form.
        return render(request, 'jerry/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        # Always return an HttpResponseRedirect after successfully dealing
        # with POST data. This prevents data from being posted twice if a
        # user hits the Back button.
        return HttpResponseRedirect(reverse('jerry:results', args=(question.id,)))
```

誰是校花

- A
- B
- C

Vote

dsjerry2017@gmail.com

CMS APP - Initial

dsjerry2017@gmail.com

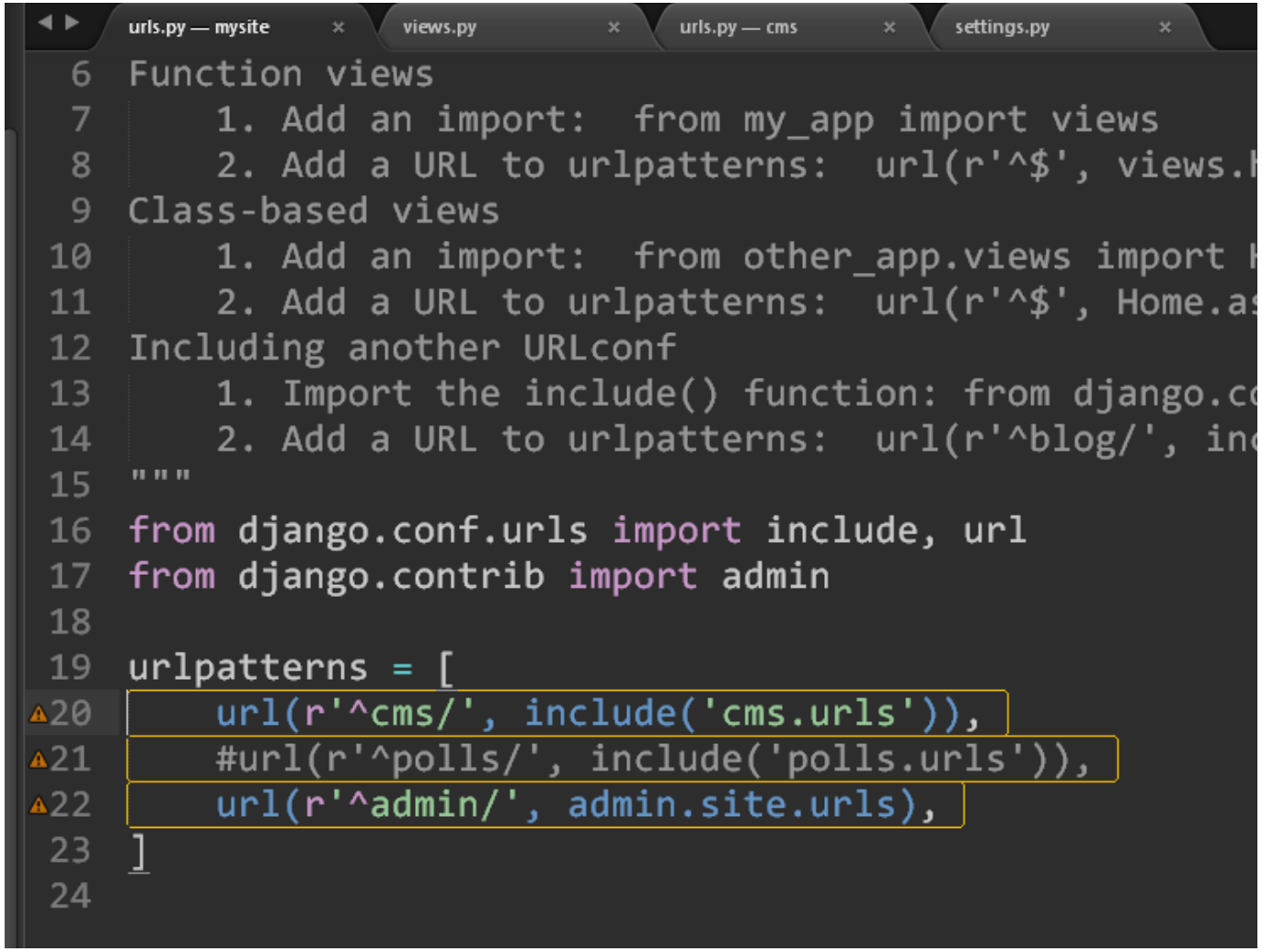
First Step

- mysite
  - settings.py
  - urls.py
- cms
  - urls.py
  - views.py

mysite/settings.py

```
# Application definition

INSTALLED_APPS = [
    'cms.apps.CmsConfig'
    'polls.apps.PollsConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
```

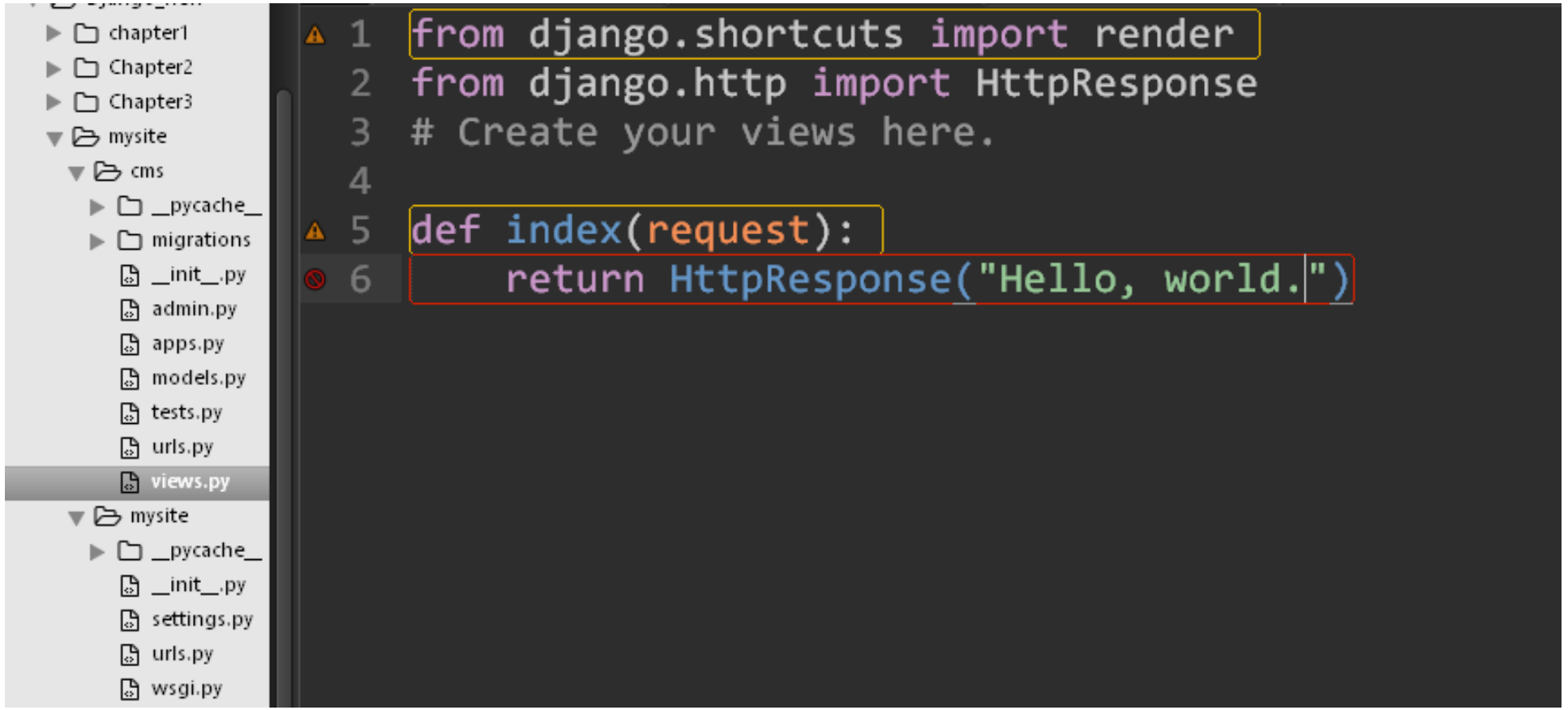
mysite/urls.py



cms/urls.py

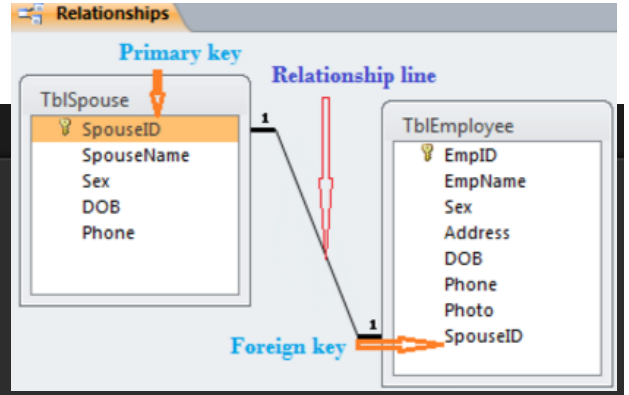

chapter1
Chapter2
Chapter3
mysite
cms
__pycache__
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
1 from django.conf.urls import url
2
3 from . import views
4
5 urlpatterns = [
6 url(r'^$', views.index, name='index'),
7]

cms/views.py


chapter1
Chapter2
Chapter3
mysite
cms
__pycache__
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
mysite
__pycache__
__init__.py
settings.py
urls.py
wsgi.py
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 # Create your views here.
4
5 def index(request):
6 return HttpResponse("Hello, world.")

CMS APP - Model

- python manage.py migrate



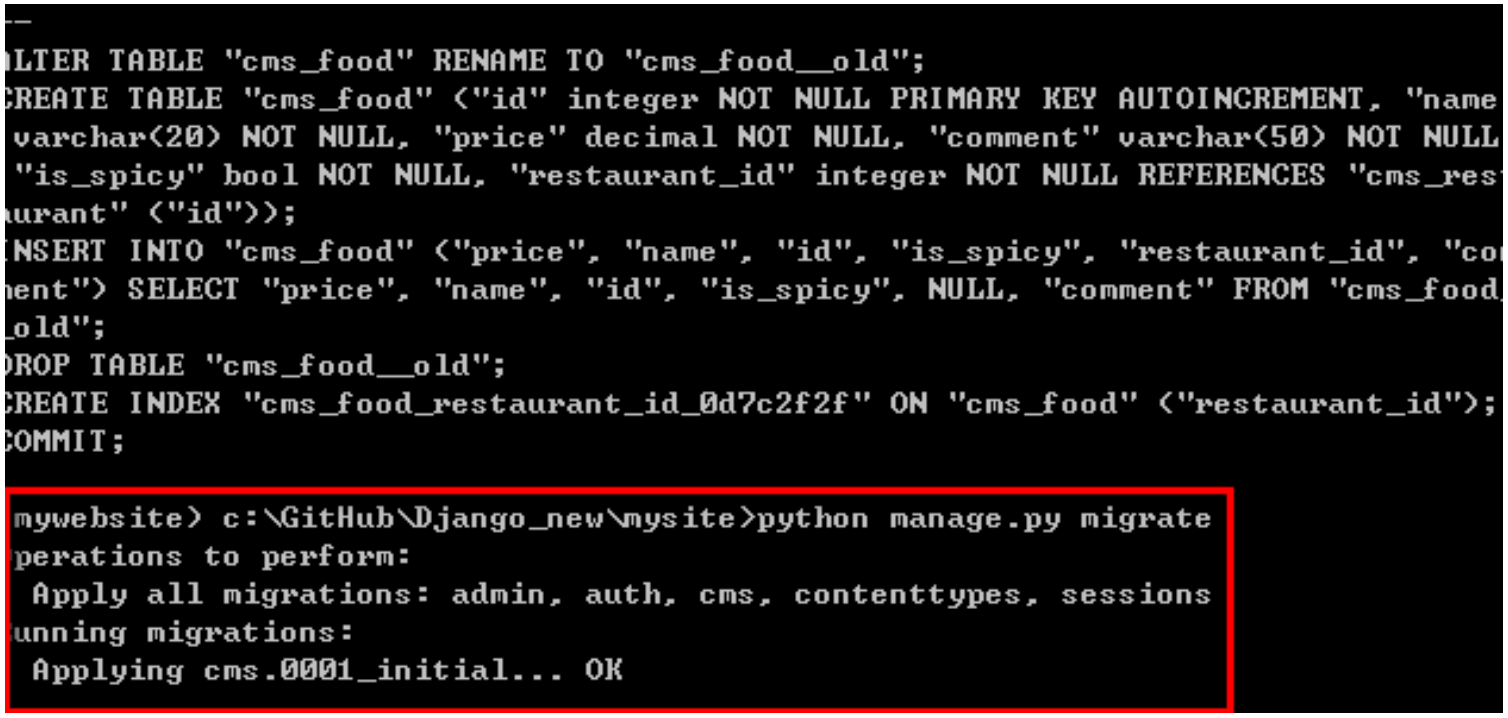
models.py

```python
from django.db import models

# Create your models here.
class Restaurant(models.Model):
    name = models.CharField(max_length=20)
    phone_number = models.CharField(max_length=15)
    address = models.CharField(max_length=50, blank=True)


class Food(models.Model):
    name = models.CharField(max_length=20)
    price = models.DecimalField(max_digits=3, decimal_places=0)
    comment = models.CharField(max_length=50, blank=True)
    is_spicy = models.BooleanField(default=False)
    restaurant = models.ForeignKey(Restaurant)
```

小數的最大位數為0

Activating models

- python manage.py makemigrations cms
- python manage.py migrate



admin.py

```
1  from django.contrib import admin
2
3  # Register your models here.
4  from .models import Restaurant,Food
5
6  admin.site.register(Food)
7  admin.site.register(Restaurant)
```

Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Cms › Foods › Add food

Add food

Name:

Price:

Comment:

Is spicy

Restaurant: ---------

Save and add another | Save and continue editing | SAVE

Select food to change

ADD FOOD +

Action: --------- Go 0 of 1 selected

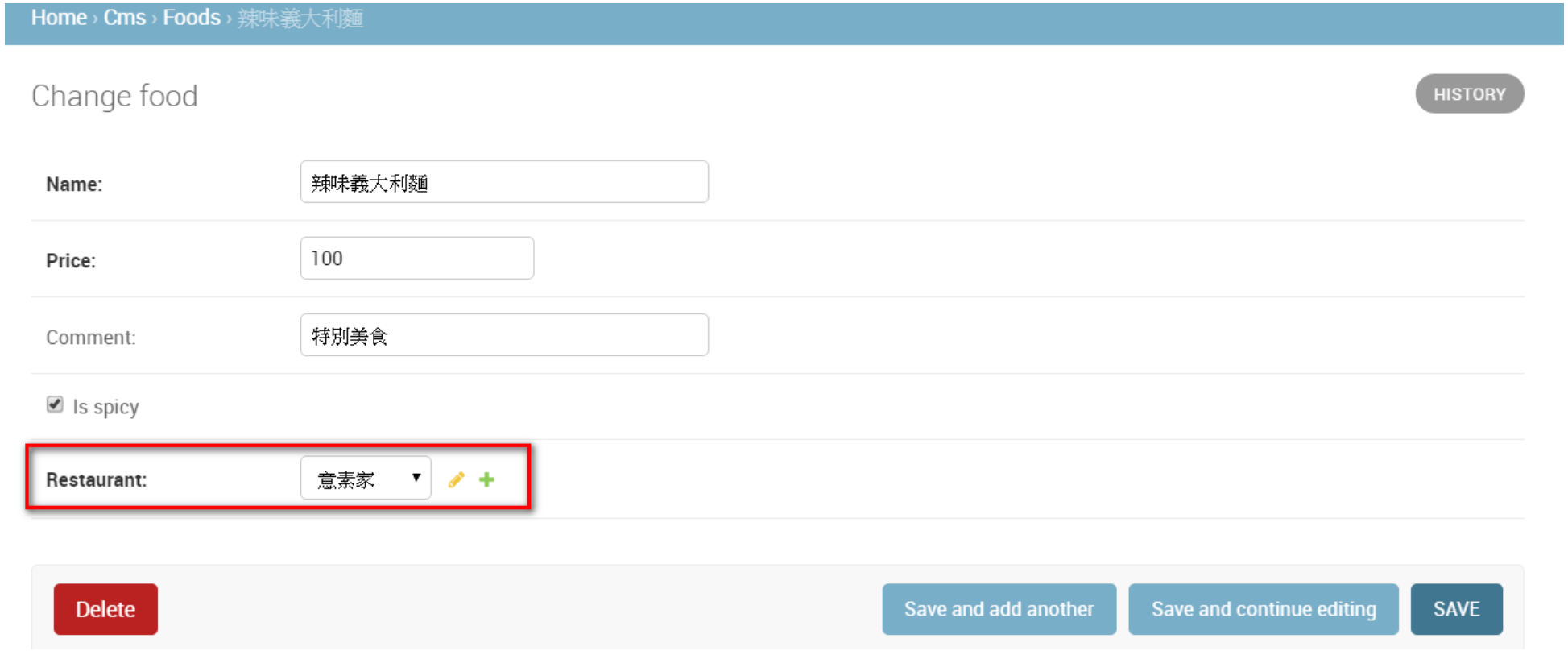
NAME	PRICE
辣味義大利麵	100

1 food

settings.py | admin.py | models.py

```python
from django.db import models

# Create your models here.
class Restaurant(models.Model):
    name = models.CharField(max_length=20)
    phone_number = models.CharField(max_length=15)
    address = models.CharField(max_length=50, blank=True)

    def __str__(self):
        return self.name

class Food(models.Model):
    name = models.CharField(max_length=20)
    price = models.DecimalField(max_digits=3, decimal_places=0)
    comment = models.CharField(max_length=50, blank=True)
    is_spicy = models.BooleanField(default=False)
    restaurant = models.ForeignKey(Restaurant)

    def __str__(self):
        return self.name
```

dsjerry2017@gmail.com



dsjerry2017@gmail.com

CMS APP - Shell

dsjerry2017@gmail.com

Display

- python manage.py shell
- from cms.models import Restaurant, Food
- restaurants=Restaurant.objects.all()
- restaurants

Create

- from cms.models import Restaurant, Food
- r1 = Restaurant.objects.create(name='台科小炒', phone_number ='02-88888', address='台北市基隆路')
- r1.save()
- Restaurant.objects.all()

```
>>> Restaurant.objects.all()
<QuerySet [<Restaurant: 意素家>, <Restaurant: 中文小館>, <Restaurant: 台科小炒>
>
```

Update

- from cms.models import Restaurant, Food
- Food.objects.all()[0].price
- Food.objects.filter(name='辣味義大利麵').update(price=200)

```
>>> Food.objects.all()[0].price
Decimal('100')
```

```
>>> Food.objects.all()[0].price
Decimal('200')
>>>
```

Delete

- from cms.models import Restaurant, Food
- Restaurant.objects.all()

- f = Restaurant.objects.get(name='台科小炒')
- f.delet

```
<QuerySet [<Restaurant: 意素家>, <Restaurant: 中文小館>, <Restaurant: 台科小炒>]
>
>>>
```

```
<QuerySet [<Restaurant: 意素家>, <Restaurant: 中文小館>]>
>>>
```

CMS APP - Views

dsjerry2017@gmail.com

Templates

```
1  from django.shortcuts import render_to_response
2  # Create your views here.
3  from .models import Restaurant, Food
4
5  def index(request):
6      restaurants = Restaurant.objects.all()
7      return render_to_response('cms/menu.html',locals())
```

```
<h1 class="text-center">美食菜單</h1>
<div class="container-fluid">
{% for r in restaurants %}
<h2>{{ r.name }}</h2>
{% if r.food_set.all %}
<p>本餐廳共有{{ r.food_set.all|length }}道菜</p>
<table class="table">
    <tr>
        <th>菜名</th>
        <th>價格</th>
        <th>辣不辣</th>
        <th>註解</th>
    </tr>
    {% for food in r.food_set.all %}
    <tr>
        <td>{{ food.name }}</td>
        <td>{{ food.price }}</td>
        <td>{% if food.is_spicy %} 辣 {% else %} 不辣 {% endif %}</td>
        <td>{{ food.comment }}</td>
    </tr>
    {% endfor %}
</table>
    {% else %}
    <p>本餐廳啥都沒賣</p>
    {% endif %}
    {% endfor %}
```

美食菜單

意素家

本餐廳共有1道菜

菜名	價格	辣不辣	註解
辣味義大利麵	200	辣	特別美食

中文小館

本餐廳啥都沒賣

dsjerry2017@gmail.com

Blog APP

dsjerry2017@gmail.com

設定APP跟時區

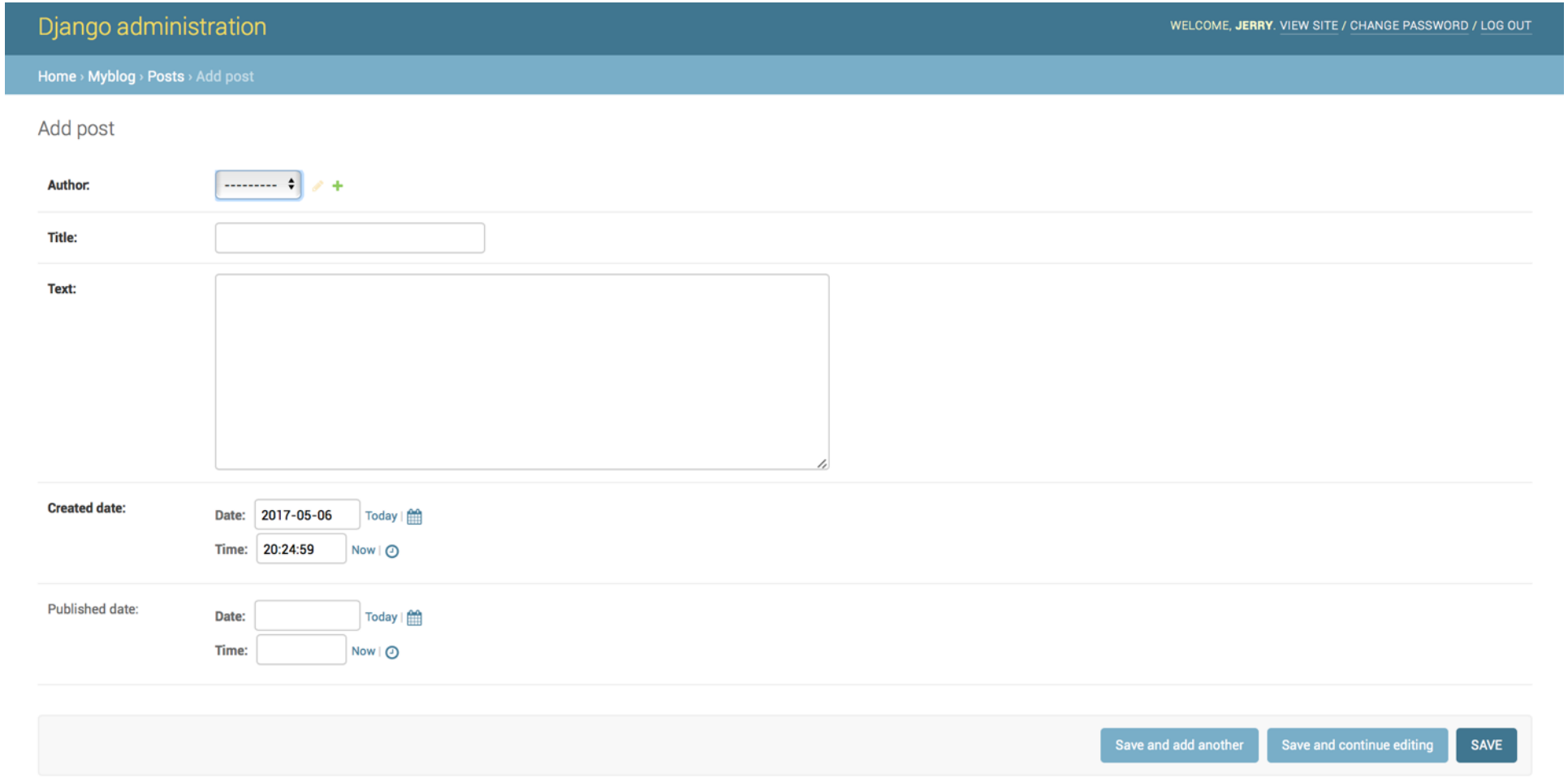
```
33  INSTALLED_APPS = [
34      'django.contrib.admin',
35      'django.contrib.auth',
36      'django.contrib.contenttypes',
37      'django.contrib.sessions',
38      'django.contrib.messages',
39      'django.contrib.staticfiles',
40      'myblog'
41  ]
42
```

建立Model

```python
# -*- coding: utf-8 -*-
from __future__ import unicode_literals

from django.db import models
from django.utils import timezone


class Post(models.Model):
    author = models.ForeignKey('auth.User')
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(
            default=timezone.now)
    published_date = models.DateTimeField(
            blank=True, null=True)

    def publish(self):
        self.published_date = timezone.now()
        self.save()

    def __str__(self):
        return self.title

```

dsjerry2017@gmail.com

建立後台


Django administration
WELCOME, JERRY. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Myblog › Posts › Add post
Add post
Author:

Title:
Text:
Created date:
Date: 2017-05-06 Today
Time: 20:24:59 Now
Published date:
Date: Today
Time: Now
Save and add another
Save and continue editing
SAVE

Project 的 urls

```
 9  Class-based views
10      1. Add an import:  from other_app.views import Home
11      2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
12  Including another URLconf
13      1. Import the include() function: from django.conf.urls import url, include
14      2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
15  """
16  from django.conf.urls import url, include
17  from django.contrib import admin
18
19  urlpatterns = [
20      url(r'^', include('myblog.urls')),
21      url(r'^admin/', admin.site.urls),
22  ]
23
```

APP的urls, views

urls

```python
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.post_list, name='post_list'),
]
```

views

```python
# -*- coding: utf-8 -*-
from __future__ import unicode_literals
from django.utils import timezone
from django.shortcuts import render
from .models import Post

# Create your views here.
def post_list(request):
    posts = Post.objects.filter(published_date__lte=timezone.now()).order_by('published_date')
    return render(request, 'blog/post_list.html', {'posts': posts})
```

建立Templates

```
1
2       {% for post in posts %}
3           <div>
4               <p>published: {{ post.published_date }}</p>
5               <h1><a href="">{{ post.title }}</a></h1>
6               <p>{{ post.text|linebreaksbr }}</p>
7           </div>
8       {% endfor %}
```

簡易Blog APP完成

published: May 6, 2017, 8:43 p.m.

將屆齡退休領18%？ 林萬億：一切按照制度

主導年金改革的行政院政委林萬億目前也是台大社工系教授，但有媒體消息指出，今年年滿65歲的林萬億已取得退休資格，將能享18%優惠存款利息，因而引發外界議論。林萬億對此則回應，「我是屆齡被強迫退休，也不是我自己要退的」，規定怎麼做就怎麼做。く

published: May 6, 2017, 8:43 p.m.

肚子餓了！ 5歲童深夜抱玩偶逛超商

新北市1名5歲的歐姓男童日前趁父母照顧弟弟睡著，深夜11時許偷溜出門，跑到住家附近的超商逗留，店員察覺有異報案，新莊所光華所員警獲報趕抵，安撫詢問後，替男童買了養樂多跟餅乾，將他帶回警局，通知父母帶回。

dsjerry2017@gmail.com

Album APP

dsjerry2017@gmail.com

設定APP與位置

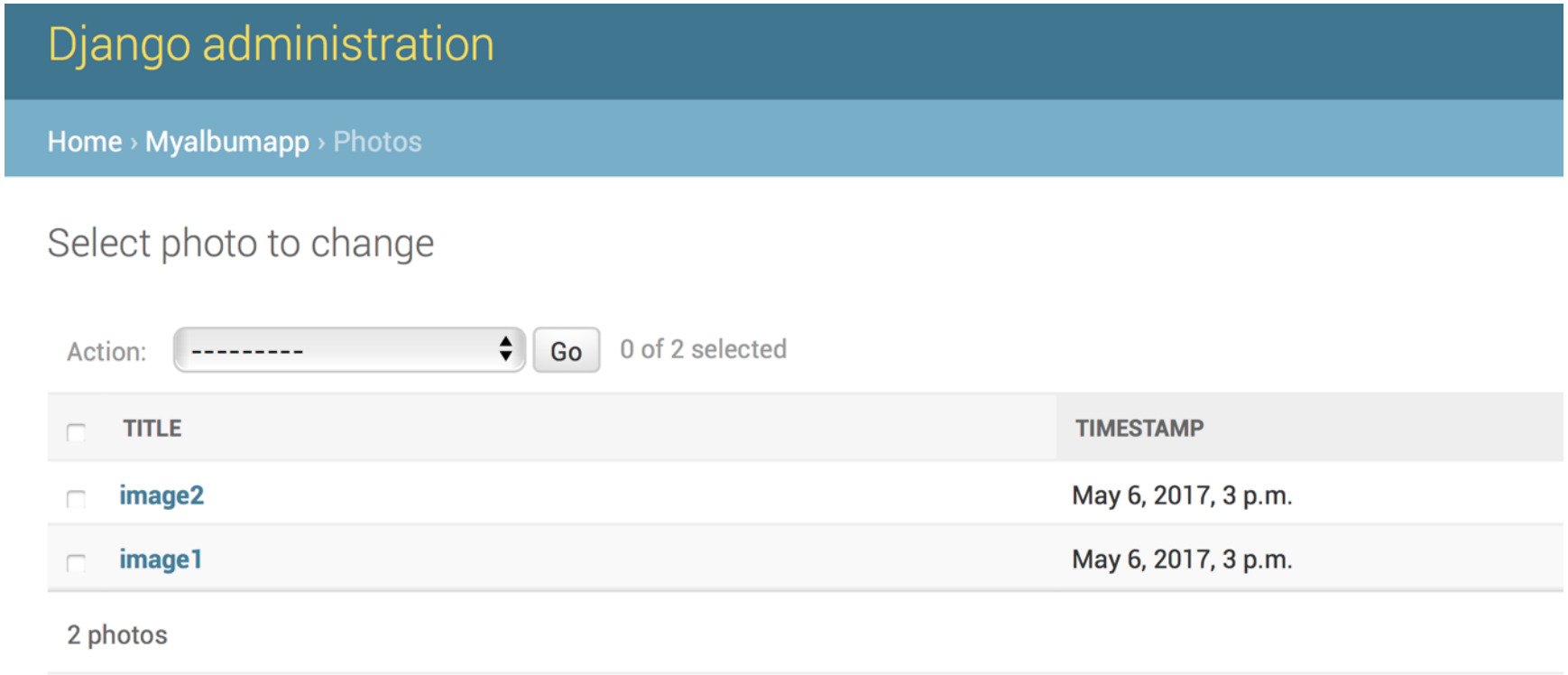
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myalbumapp'
]
```

```
MEDIA_URL = '/media/'
MIDIA_ROOT = os.path.join(os.path.dirname(BASE_DIR), "mideafiles")
```

建立Model

```python
# -*- coding: utf-8 -*-
from __future__ import unicode_literals
from django.db import models

class Photo(models.Model):
    title = models.CharField(max_length=200)
    width = models.IntegerField(default=0)
    height = models.IntegerField(default=0)
    image = models.ImageField(null=False, blank=False, width_field="width", height_field="height")
    timestamp = models.DateTimeField(auto_now_add=True, auto_now=False)

    def __str__(self):
        return self.title

    class Meta:
        ordering = ["-timestamp"]
```

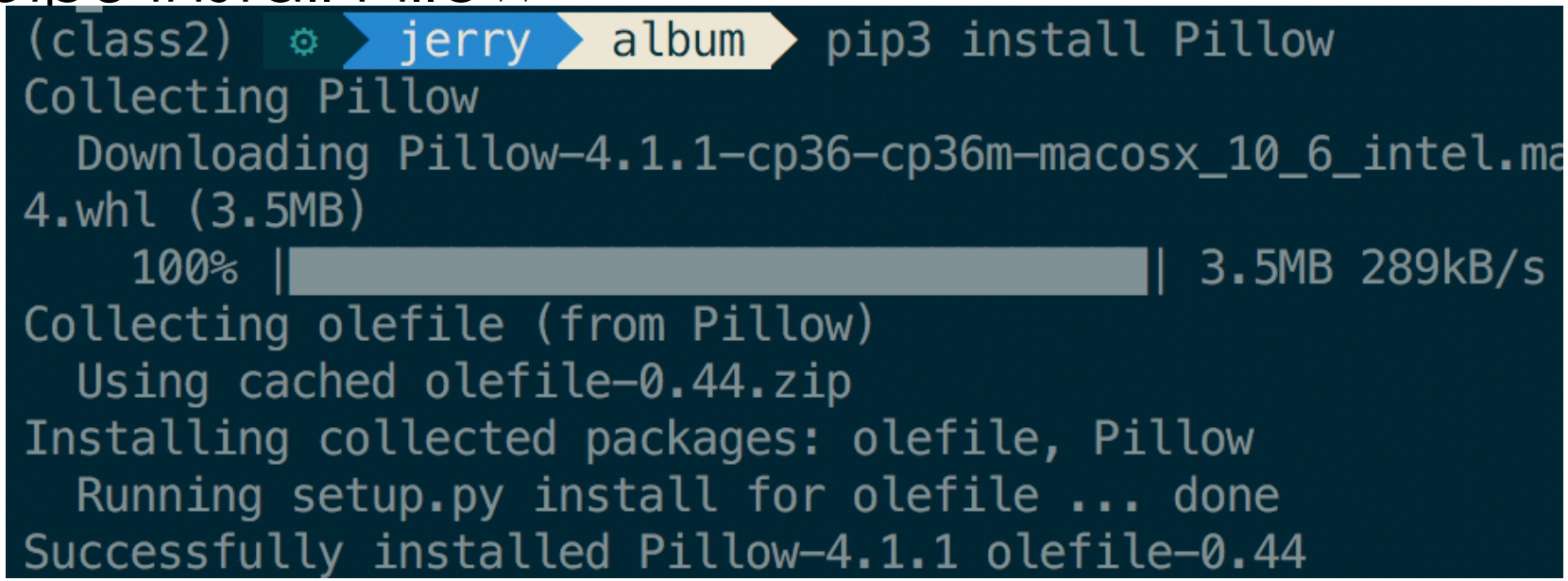
建立後台


Django administration
Home › Myalbumapp › Photos
Select photo to change
Action:

Go
0 of 2 selected
TITLE
TIMESTAMP
image2
May 6, 2017, 3 p.m.
image1
May 6, 2017, 3 p.m.
2 photos

安裝Python 圖片工具

- 安裝pip3 install Pillow



Project 的 urls

```
from django.conf import settings
from django.conf.urls.static import static
from django.conf.urls import url, include
from django.contrib import admin


urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^', include('myalbumapp.urls')),
]

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

APP的urls, views

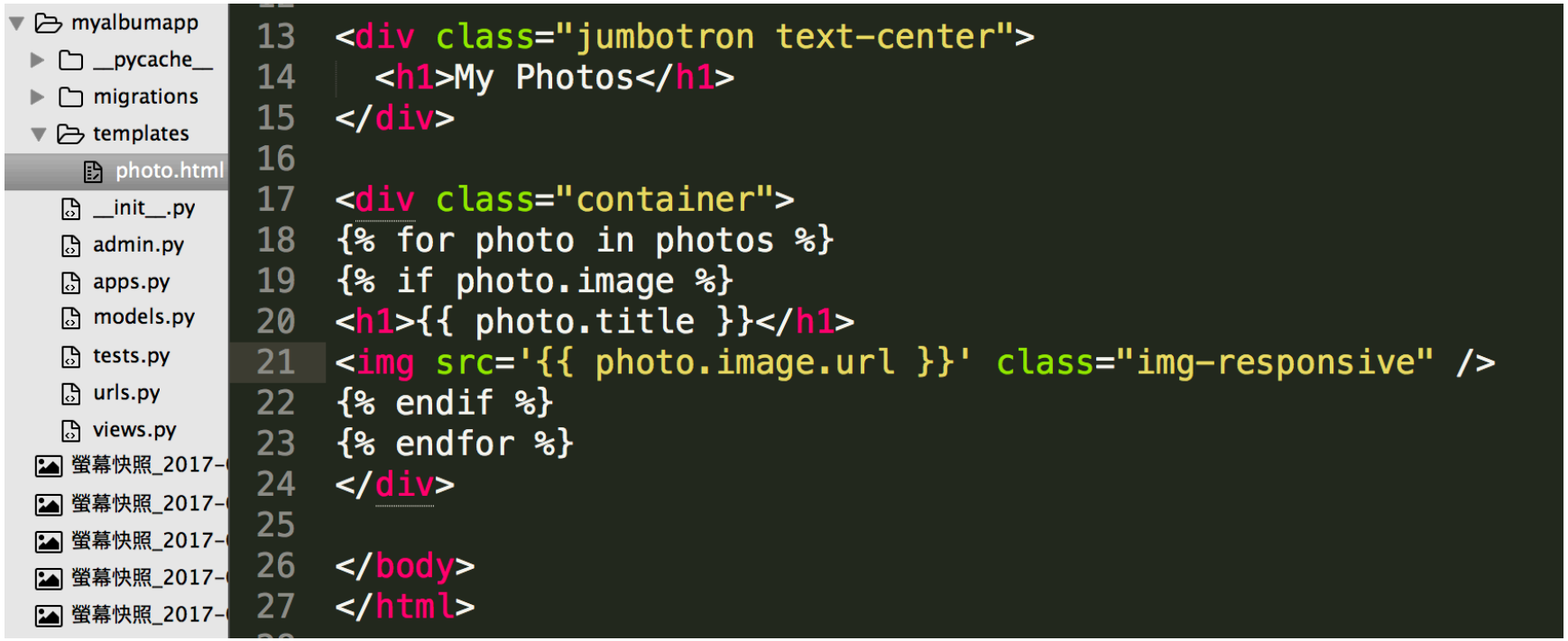
urls

```python
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.photo_list, name='photo_list'),
]
```

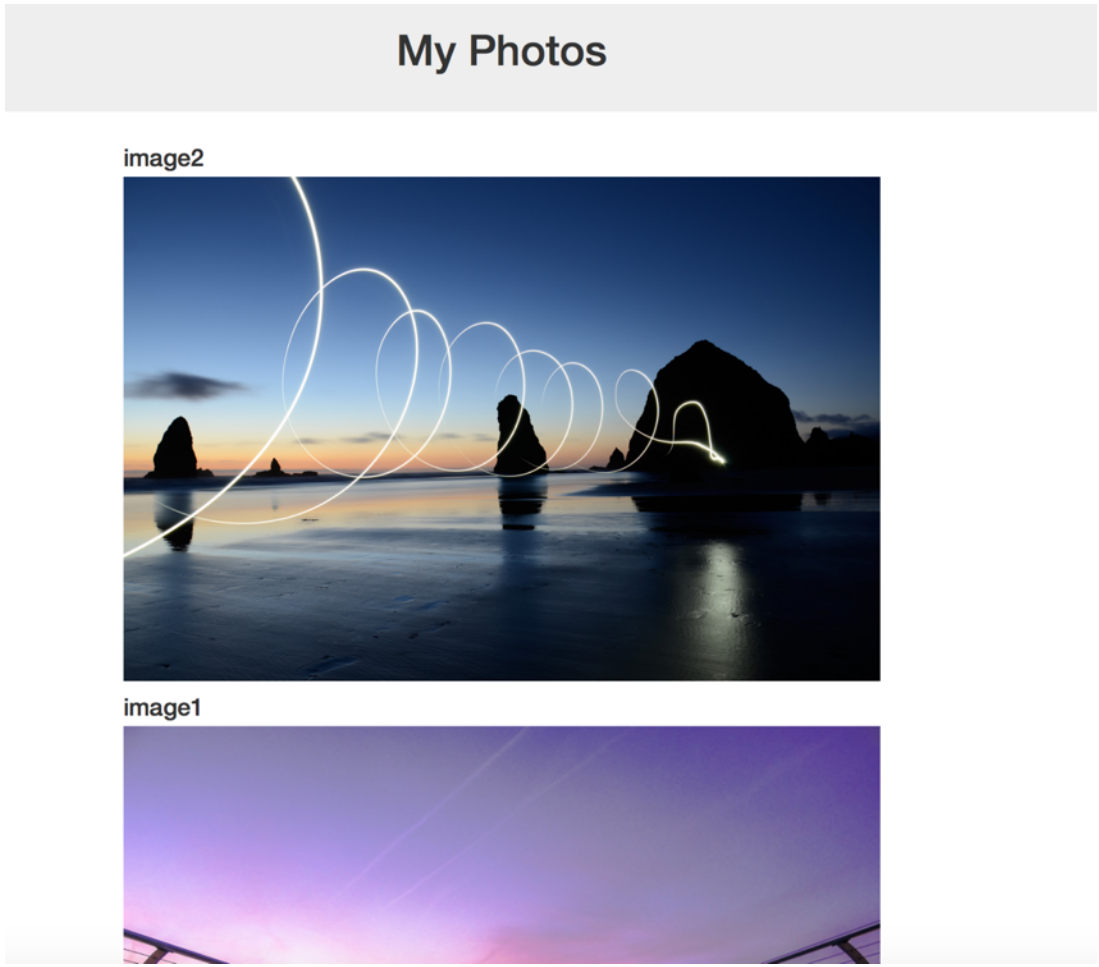
views

```python
from __future__ import unicode_literals

from django.shortcuts import render
from .models import Photo
# Create your views here.

def photo_list(request):
    queryset = Photo.objects.all()
    context = {
        "photos": queryset,
    }
    return render(request, "photo.html", context)
```

建立Templates



成果



dsjerry2017@gmail.com

dsjerry2017@gmail.com
jerry@mail.ntust.edu.tw