Part4

更新投票細節的模板，增加一個包含html<form>的元素

polls/templates/polls/detail.html：

<h1>{{ question.question\_text }}</h1>

{% if error\_message %}<p><strong>{{ error\_message }}</strong></p>{% endif %}

<form action="{% url 'polls:vote' question.id %}" method="post">

{% csrf\_token %}

{% for choice in question.choice\_set.all %}

<input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />

<label for="choice{{ forloop.counter }}">{{ choice.choice\_text }}</label><br />

{% endfor %}

<input type="submit" value="Vote" />

</form>

之前創建了vote()函數，現在創一個真正的版本。將以下內容添加到polls/views.py

polls/views.py：

from django.shortcuts import get\_object\_or\_404, render

from django.http import HttpResponseRedirect, HttpResponse

from django.urls import reverse

from .models import Choice, 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, '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,)))

有人在問題中投票後，vote()會統計

polls/views.py：

from django.shortcuts import get\_object\_or\_404, render

def results(request, question\_id):

question = get\_object\_or\_404(Question, pk=question\_id)

return render(request, 'polls/results.html', {'question': question})

在polls/templates/polls裡新增一個results.html的檔案

polls/templates/polls/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>

之後開啟polls/urls.py新增並更改code

from django.urls import path

from . import views

app\_name = 'polls'

urlpatterns = [

path('', views.IndexView.as\_view(), name='index'),

path('<int:pk>/', views.DetailView.as\_view(), name='detail'),

path('<int:pk>/results/', views.ResultsView.as\_view(), name='results'),

path('<int:question\_id>/vote/', views.vote, name='vote'),

]

刪除舊的索引，打開polls/views.py更改code

from django.shortcuts import get\_object\_or\_404, render

from django.http import HttpResponseRedirect

from django.urls import reverse

from django.views import generic

from .models import Choice, Question

class IndexView(generic.ListView):

template\_name = 'polls/index.html'

context\_object\_name = 'latest\_question\_list'

def get\_queryset(self):

"""Return the last five published questions."""

return Question.objects.order\_by('-pub\_date')[:5]

class DetailView(generic.DetailView):

model = Question

template\_name = 'polls/detail.html'

class ResultsView(generic.DetailView):

model = Question

template\_name = 'polls/results.html'

def vote(request, question\_id):

... # same as above, no changes needed.

Part6

在polls裡建立static的資料夾，之後再static裡建立polls的資料夾，最後在polls裡建立style.css的文件

polls/static/polls/style.css：

li a {

color: green;

}

之後增加以下的code到polls/templates/polls/index.html

{% load static %}

<link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}" />

這段code是去產生絕對URL的檔案

新增樣式表

polls/static/polls/style.css：

body {

background: white url("images/background.gif") no-repeat;

}

之後到terminal裡輸入http://localhost:8000/polls/ 去看螢幕左上角有沒有加載的背景

Part7

去polls/admin.py全部複製貼上

127.0.0.1:8000/admin

去polls/admin.py全部複製貼上

去polls/admin.py 改前兩句

**from** **django.contrib** **import** admin

**from** **.models** **import** Choice, Question

倒數第二行加

admin.site.register(Choice)

去polls/admin.py加

**class** **ChoiceInline**(admin.StackedInline):

model = Choice

extra = 3

**class** **QuestionAdmin**(admin.ModelAdmin):

fieldsets = [

(**None**, {'fields': ['question\_text']}),

('Date information', {'fields': ['pub\_date'], 'classes': ['collapse']}),

]

inlines = [ChoiceInline]

去polls/admin.py改

**class** **ChoiceInline**(admin.TabularInline):

去polls/admin.py加上

list\_display = ('question\_text', 'pub\_date')

修改

list\_display = ('question\_text', 'pub\_date', 'was\_published\_recently')

到去 polls/models.py 改

**def** was\_published\_recently(self):

now = timezone.now()

**return** now - datetime.timedelta(days=1) <= self.pub\_date <= now

was\_published\_recently.admin\_order\_field = 'pub\_date'

was\_published\_recently.boolean = **True**

was\_published\_recently.short\_description = 'Published recently?'

去terminal

$python manage.py makemigrations

$python manage.py migrate

$python manage.py runserver

polls/admin.py QuestionAdmin:加上

list\_filter = ['pub\_date']

再加上

search\_fields = ['question\_text']

去mysite/settings.py 改這段

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

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

'APP\_DIRS': **True**,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

在mysite的目錄下開一個templates資料夾跟manage.py一樣位置

之後在templates裡面開一個資料夾admin

點VENV搜尋 Django (左上第一個)

django/contrib/admin/templates/admin 複製base\_site.html 檔案

貼到剛剛創的admin目錄

編輯 貼第三段

{% **block** branding %}

<**h1** id="site-name"><**a** href="{% **url** 'admin:index' %}">Polls Administration</**a**></**h1**>

{% **endblock** %}

變成(Polls Administration)