**0643010王儷蓉，0643011楊喻晴，0643024施欣佑**

**0643033潘玟晴，0643046洪于婷，0643047劉潔明**

Part3-2

* 在polls/views.py裡的def index(request)刪除原本的貼上:

def index(request):

latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]

context = {'latest\_question\_list': latest\_question\_list}

return render(request, 'polls/index.html', context)

* 在polls/views.py裡刪除原本的貼上:

from django.shortcuts import render

from .models import Question

* >>>在網址輸入127.0.0.1:8000/polls/會出現可連結的What’s up?
* 在polls/views.py裡刪除原本的貼上:

from django.http import Http404

from django.shortcuts import render

from .models import Question

* 在polls/views.py裡的def index(request)刪除原本的貼上:

def detail(request, question\_id):

try:

question = Question.objects.get(pk=question\_id)

except Question.DoesNotExist:

raise Http404("Question does not exist")

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

* >>>到polls資料夾里再開一個名為detail.html的文件檔並貼上

{{ question }}

* 在polls/views.py裡的def index(request)刪除原本的貼上:

def detail(request, question\_id):

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

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

* 在polls/views.py裡刪除原本的貼上:

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

from .models import Question

* 在polls/templates/polls/detail.html裡刪除原本的貼上:

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

<ul>

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

<li>{{ choice.choice\_text }}</li>

{% endfor %}

</ul>

* >>>在網址輸入127.0.0.1:8000/polls/1/會出現下方文字

What’s up?

* Not much
* The sky
* 在polls/urls.py裡刪除原本的貼上:

from django.urls import path

from . import views

app\_name = 'polls'

urlpatterns = [

path('', views.index, name='index'),

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

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

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

]

* 在polls/templates/polls/index.html裡

刪除

{% for question in latest\_question\_list %}

<li><a href="/polls/{{ question.id }}/">{{ question.question\_text }}</a></li>

貼上

<li><a href="{% url 'detail' question.id %}">{{ question.question\_text }}</a></li>

* 在polls/templates/polls/index.html裡

刪除

<li><a href="{% url 'detail' question.id %}">{{ question.question\_text }}</a></li>

貼上

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

Part4

* 在polls/templates/polls/detail.html裡面刪掉並貼上

<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>

* polls/views.py

刪除

from django.shortcuts import render

from .models import Question

新增

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

刪除：

def vote(request, question\_id):

return HttpResponse("You're voting on question %s." % question\_id)

新增：

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,)))

* 在polls/views.py

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

刪除：

def results(request, question\_id):

response = "You're looking at the results of question %s."

return HttpResponse(response % question\_id)

新增：

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新增results.html

貼上

<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>

* 在polls/templates/polls中建立一個result.html貼上

<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>

* 在polls/views.py刪除原本的貼上

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

* 在polls/urls.py中刪除原本的貼上

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'),

]

* 在polls/views.py中

刪除

def index(request):

latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]

context = {'latest\_question\_list': latest\_question\_list}

return render(request, 'polls/index.html', context)

def detail(request, question\_id):

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

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

def results(request, question\_id):

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

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

貼上

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'

* 在polls/urls.py中最後加上

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

* polls/static/polls/style.css

在polls資料夾裡面新增一個資料夾叫static, 然後再在static裡面新增一個資料夾叫polls, 然後在polls裡面新增style.css文件

* 在style.css裡面貼

**li** **a** {

**color**: **green**;

}

讓what’s up 字體呈現綠色

* polls/templates/polls/index.html

在index.html裡面的最上面“加”上

{% **load** static %}

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

讓index.html去下載static的資料

* **polls/static/polls/images/background.gif**

然後在static裡的polls中新增一個資料夾叫images,最後上網下載一個圖片改名叫b**ackground.gif的圖片,** 副檔名可以改jpg等等,然後將圖片移到images的資料夾裡面

* polls/static/polls/style.css

在style.css裡面加上

**body** {

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

}

使[h](http://localhost:8000/polls/)**[ttp://localhost:8000/polls/](http://localhost:8000/polls/)** 的網頁背景呈現你剛剛下載在images裡面的圖片

用ubutun 應該是 http://127.0.0.1:8000/polls/

如果照片名稱副檔名有改jpg等等的要在"images/background.gif"改成"images/background.jpg"

Part7

* 在polls/admin.py裡刪除原本的貼上:

from django.contrib import admin

from .models import Question

class QuestionAdmin(admin.ModelAdmin):

fields = ['pub\_date', 'question\_text']

admin.site.register(Question, QuestionAdmin)

* 搜尋

http://127.0.0.1:8000/admin/

<http://127.0.0.1:8000/polls/1/>

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

<http://127.0.0.1:8000/polls/1/vote/>

* 在polls/admin.py裡刪除原本的貼上:

from django.contrib import admin

from .models import Question

class QuestionAdmin(admin.ModelAdmin):

fieldsets = [

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

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

]

admin.site.register(Question, QuestionAdmin)

* 在polls/admin.py最下面新增:

admin.site.register(Choice)

* 在polls/admin.py

刪除

class QuestionAdmin(admin.ModelAdmin):

fieldsets = [

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

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

]

增加:

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]

* 在polls/admin.py

刪除

class ChoiceInline(admin.StackedInline):

增加

class ChoiceInline(admin.TabularInline):

* 在polls/admin.py

的class QuestionAdmin(admin.ModelAdmin): 對齊最後一行 新增

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

如

class QuestionAdmin(admin.ModelAdmin):

fieldsets = [

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

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

]

inlines = [ChoiceInline]

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

* 在polls/admin.py

將上一個新增的 list\_display = ('question\_text', 'pub\_date') 刪除

新增

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

* 在polls/models.py

刪除

def was\_published\_recently(self):

return self.pub\_date >= timezone.now() - datetime.timedelta(days=1)

新增

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?'

**害怕錯的話 執行以下三個指令**

* $ python manage.py makemigrations 創建更改的文件
* $ python manage.py migrate將資料庫做更新
* $ python manage.py runserver 啟動開發伺服器
* 在polls/admin.py裡 新增

from django.contrib import admin

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', 'was\_published\_recently')

**list\_filter = ['pub\_date']**

admin.site.register(Question, QuestionAdmin)

admin.site.register(Choice)

* 在剛剛新增的**list\_filter = ['pub\_date']** 的下面 新增

**search\_fields = ['question\_text']**

* 在mysite/settings.py

新增

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templates資料夾裡面新增一個admin資料夾

再新增base\_site.html

打開後貼上

{% block branding %}

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

{% endblock %}