

LAB:10

Student Task

Create a small app containing all concept which we discuss in lab 9 & 10

```
1 from django.http import HttpResponseRedirect, HttpResponseRedirect
2 from django.template import loader
3 from django.shortcuts import get_object_or_404, render
4 from django.views import generic
5 from django.utils import timezone
6 from django.urls import reverse
7 from .models import Choice, Question
8
9 def index(request):
10     latest_question_list = Question.objects.order_by('-pub_date')[:5]
11     template = loader.get_template('polls/index.html')
12     context = {
13         'latest_question_list': latest_question_list,
14     }
15     return HttpResponse(template.render(context, request))
16
17     #output = ', '.join([q.question_text for q in latest_question_list])
18     #return HttpResponse(output)
19     #return HttpResponse("Hello, world. You're at the polls index.")
20
21
22
23 class IndexView(generic.ListView):
24     template_name = 'polls/index.html'
25     context_object_name = 'latest_question_list'
26
27     def get_queryset(self):
28         return Question.objects.filter(pub_date__lte=timezone.now()).order_by('-pub_date')[:5]
```

Figure 1 Views.py

```
32 class DetailView(generic.DetailView):
33     model = Question
34     template_name = 'polls/detail.html'
35     def get_queryset(self):
36         """
37         Excludes any questions that aren't published yet.
38         """
39         return Question.objects.filter(pub_date__lte=timezone.now())
40
41
42 class ResultsView(generic.DetailView):
43     model = Question
44     template_name = 'polls/results.html'
45
46
47
48 def detail(request, question_id):
49     try:
50         question = Question.objects.get(pk=question_id)
51     except Question.DoesNotExist:
52         raise Http404("Question does not exist")
53     return render(request, 'polls/detail.html', {'question': question})
54
55     #return HttpResponse("You are looking at question %s." % question_id)
56
57 def results(request, question_id):
58     question = get_object_or_404(Question, pk=question_id)
59     return render(request, 'polls/results.html', {'question': question})
60
```

Figure 2 Views.py

```
polls\urls.py x mysite\urls.py x apps.py x admin.py x settings.py x detail.html x
1 from django.urls import path
2
3 from . import views
4 app_name = 'polls'
5 urlpatterns = [
6     path('', views.IndexView.as_view(), name='index'),
7     path('<int:pk>/', views.DetailView.as_view(), name='detail'),
8     path('<int:pk>/results/', views.ResultsView.as_view(), name='results'),
9     path('<int:question_id>/vote/', views.vote, name='vote'),
10 ]
11
```

Figure 3 polls/urls.py

```
polls\urls.py x mysite\urls.py x apps.py x admin.
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('', include('polls.urls')),
6     path('polls/', include('polls.urls')),
7     path('admin/', admin.site.urls),
8 ]
9
```

Figure 4 Urls.py (mysite)

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

Figure 5 Details.html

```

1  from django.db import models
2      import datetime
3  from django.utils import timezone
4
5  class Question(models.Model):
6      question_text = models.CharField(max_length=200)
7      pub_date = models.DateTimeField('date published')
8
9      def __str__(self):
10         return self.question_text
11
12     def was_published_recently(self):
13         now = timezone.now()
14         return now - datetime.timedelta(days=1) <= self.pub_date <= now
15         #return self.pub_date >= timezone.now() - datetime.timedelta(days=1)
16     was_published_recently.admin_order_field = 'pub_date'
17     was_published_recently.boolean = True
18     was_published_recently.short_description = 'Published recently?'
19
20 class Choice(models.Model):
21     question = models.ForeignKey(Question, on_delete=models.CASCADE)
22     choice_text = models.CharField(max_length=200)
23     votes = models.IntegerField(default=0)
24     def __str__(self):
25         return self.choice_text

```

Figure 6 Models.py

```

1  {% load static %}
2
3  <link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}">
4  {% if latest_question_list %}
5      <ul>
6          {% for question in latest_question_list %}
7              <li><a href="{% url 'polls:detail' question.id %}">{{ question.question_text }}</a></li>
8              {% endfor %}
9          </ul>
10 {% else %}
11     <p>No polls are available.</p>
12 {% endif %}

```

Figure 7 Index.html

```

1  <h1>{{ question.question_text }}</h1>
2
3  <ul>
4      {% for choice in question.choice_set.all %}
5          <li>{{ choice.choice_text }} -- {{ choice.votes }} vote{{ choice.votes|pluralize }}</li>
6          {% endfor %}
7      </ul>
8
9  <a href="{% url 'polls:detail' question.id %}">Vote again?</a>

```

Figure 8 Results.html

Output:

What's up?

- ☐ Not much
- ☐ The Sky
- ☐ Just Hacking Again!

What's up?

- Not much -- 1 vote
- The Sky -- 2 votes
- Just Hacking Again! -- 6 votes

[Vote again?](#)

-
- [What's up?](#)
 - [Most Popular Gaming Channel on youtube?](#)
 - [Best cricket player?](#)

Best Cricket player?

- ☐ Virat Kohli
- ☒ Babar Azam
- ☐ Shanne Watson

Best Cricket player?

- Virat Kohli -- 2 votes
- Babar Azam -- 11 votes
- Shanne Watson -- 1 vote

[Vote again?](#)