# LAB 9-10

**Polling App**

Index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title> Poll App</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">  
  
</head>  
<body>  
<div class="p-3 mb-2 bg-warning text-dark">  
  
  
  
{% if latest\_question\_list %}  
 <ul>  
 {% for question in latest\_question\_list %}  
 <li><a href="/poll/{{ question.id }}/">  
 {{ question.question\_text }}</a></li>  
 {% endfor %}  
 </ul>  
{% else %}  
 <p>No polls are available.</p>  
{% endif %}  
  
  
  
  
</div>  
</body>  
</html>

Polls/urls.py

from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('poll/', include('poll.urls', namespace='poll')),  
  
]

Mysite/urls.py

from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('poll/', include('poll.urls', namespace='poll')),  
  
]

Views.py

from django.shortcuts import render, get\_object\_or\_404  
from django.http import HttpResponse, Http404, HttpResponseRedirect  
from django.template import loader  
from .models import Question, Choice  
from django.urls import reverse  
  
  
# Create your views here.  
  
def index(request):  
 latest\_question\_list = Question.objects.order\_by('-pub\_date')[:5]  
 template = loader.get\_template('poll/index.html')  
 context = {  
 'latest\_question\_list': latest\_question\_list,  
 }  
 return HttpResponse(template.render(context, request))  
  
  
def details(request, questions\_id):  
 try:  
 question = Question.objects.get(pk=questions\_id)  
 except Question.DoesNotExist:  
 raise Http404("This question doesnot exists.")  
 return render(request, 'poll/details.html', {'question': question})  
  
  
def results(request, questions\_id):  
 question = get\_object\_or\_404(Question, pk=questions\_id)  
 return render(request, 'poll/results.html', {'question': question})  
  
  
def vote(request, questions\_id):  
 question = get\_object\_or\_404(Question, pk=questions\_id)  
 try:  
 selected\_choice = question.choice\_set.get(pk=request.POST['choice'])  
 except (KeyError, Choice.DoesNotExist):  
 # Redisplay the question voting form.  
 return render(request, 'poll/details.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('poll:results', args=(question.id,)))

