**Web framework practical assignment -18**

Develop a basicpoll application(app).It should consist of twoparts:

a)  A publicsite inwhich user can pick theirfavourite programming language and vote.

b)    Anadmin site that lets youadd, changeand deleteprogramming languages

Ans-

Create project name poll\_app

Then create to app inside that polls and pages

Polls-model.py-

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

polls-admin.py

from django.contrib import admin

# Register your models here.

from .models import Question, Choice

# admin.site.register(Question)

# admin.site.register(Choice)

admin.site.site\_header = "Pollster Admin"

admin.site.site\_title = "Pollster Admin Area"

admin.site.index\_title = "Welcome to the Pollster Admin Area"

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]

admin.site.register(Question, QuestionAdmin)

polls->views.py

from django.template import loader

from django.http import HttpResponse, HttpResponseRedirect

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

from django.urls import reverse

from .models import Question, Choice

# Get questions and display them

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)

# Show specific question and choices

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})

# Get question and display results

def results(request, question\_id):

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

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

# Vote for a question choice

def vote(request, question\_id):

    # print(request.POST['choice'])

    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, )))

polls->urls.py

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'),

]

Polls->templates->polls->index.html

{% extends 'base.html' %}

{% block content %}

<h1 class ="text-center mb-3">Poll Questions</h1>

{% if latest\_question\_list %}

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

<div class ="card-mb-3">

    <div class ="card-body">

        <p class ="lead">{{ question.question\_text }}</p>

        <a href ="{% url 'polls:detail' question.id %}" class ="btn btn-primary btn-sm">Vote Now</a>

        <a href ="{% url 'polls:results' question.id %}" class ="btn btn-secondary btn-sm">Results</a>

    </div>

</div>

{% endfor %}

{% else %}

<p>No polls available</p>

{% endif %}

{% endblock %}

Polls->templates->polls->result.html

{% extends 'base.html' %}

{% block content %}

<h1 class ="mb-5 text-center">{{ question.question\_text }}</h1>

<ul class ="list-group mb-5">

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

    <li class ="list-group-item">

        {{ choice.choice\_text }} <span class ="badge badge-success float-right">{{ choice.votes }}

            vote{{ choice.votes | pluralize }}</span>

    </li>

    {% endfor %}

</ul>

<a class ="btn btn-secondary" href ="{% url 'polls:index' %}">Back To Polls</a>

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

{% endblock %}

Polls->templates->polls->detail.html

{% extends 'base.html' %}

{% block content %}

<a class ="btn btn-secondary btn-sm mb-3" href ="{% url 'polls:index' %}">Back To Polls</a>

<h1 class ="text-center mb-3">{{ question.question\_text }}</h1>

{% if error\_message %}

<p class ="alert alert-danger">

    <strong>{{ error\_message }}</strong>

</p>

{% endif %}

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

    {% csrf\_token %}

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

    <div class ="form-check">

        <input type ="radio" name ="choice" class ="form-check-input" id ="choice{{ forloop.counter }}"

            value ="{{ choice.id }}" />

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

    </div>

    {% endfor %}

    <input type ="submit" value ="Vote" class ="btn btn-success btn-lg btn-block mt-4" />

</form>

{% endblock %}

Polls->templates->base.html

<! DOCTYPE html>

<html lang ="en">

<head>

    <title>Pollster {% block title %}{% endblock %}</title>

</head>

<body>

    <!--NavBar-->

    {% include 'partials/\_navbar.html'%}

    <div class ="container">

        <div class ="row">

            <div class =".col-md-6 m-auto">

                {% block content %}{% endblock %}

            </div>

        </div>

    </div>

</body>

</html>

Pages->views.py

from django.shortcuts import render

# Create your views here.

def index(request):

    return render(request, 'pages/index.html')

pages->urls.py

from django.urls import path

from . import views

urlpatterns = [

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

]

Pages->templates->pages->index.html

{% extends 'base.html' %}

{% block content %}

<div class ="card text-center">

    <div class ="card-body">

        <h1>Welcome To Pollster</h1>

        <p>This is an example Django polling app</p>

        <a class ="btn btn-dark" href ="{% url 'polls:index' %}">

            View Available Polls</a>

    </div>

</div>

{% endblock %}

Main url.py

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

    path('', include('pages.urls')),

    path('polls/', include('polls.urls')),

    path('admin/', admin.site.urls),

]

Output-