Pollme/Settings.py:

```
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'x*za6xf&_80ofdpae!yzq61g9ffikkx9$*iygbl$j7rr4wlf8t'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = False

ALLOWED_HOSTS = ['*']

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'polls.apps.PollsConfig',
    'accounts.apps.AccountsConfig',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'polls.middleware.StateMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'django_session_timeout.middleware.SessionTimeoutMiddleware',
]

ROOT_URLCONF = 'pollme.urls'

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',
            ],
        },
    },
]

WSGI_APPLICATION = 'pollme.wsgi.application'


# Database

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Set whether the session cookies should be secure (HTTPS-only)
```

```
SESSION_COOKIE_SECURE = True

# Set whether the session cookies should be HTTP only
SESSION_COOKIE_HTTPONLY = True

#Session Timeout
SESSION_EXPIRE_SECONDS = 120 # 2 min in seconds

SESSION_EXPIRE_AT_BROWSER_CLOSE=True # Invalidate session


# Internationalization

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Africa/Nairobi'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, Images)

STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static')
]
```

pollme/urls.py:

```
from django.contrib import admin
from django.urls import path, include
from . import views


urlpatterns = [
    path('admin/', admin.site.urls),
    path('',  views.login_user, name='login'),
    path('home/', views.home, name='home'),
    path('polls/', include('polls.urls', namespace="polls")),
    path('accounts/', include('accounts.urls', namespace="accounts")),
]
```

Pollme/views.py:

```
from django.shortcuts import render
from accounts.views import login_user

def home(request):
    return render(request,'home.html')
```

then an app called accounts:

accounts/apps.py:

```
from django.apps import AppConfig


class AccountsConfig(AppConfig):
    name = 'accounts'
```

accounts/forms.py:

```
from django import forms
from django.contrib.auth.models import User
```

```
class UserRegistrationForm(forms.Form):
    username = forms.CharField(label='Username', max_length=100, min_length=5,
                               widget=forms.TextInput(attrs={'class': 'form-
control'}))
    email = forms.EmailField(label='Email', max_length=20, min_length=5,
                             widget=forms.EmailInput(attrs={'class': 'form-
control'}))
    password1 = forms.CharField(label='Password', max_length=50, min_length=5,
                                widget=forms.PasswordInput(attrs={'class': 'form-
control'}))
    password2 = forms.CharField(label='Confirm Password',
                                max_length=50, min_length=5,
                                widget=forms.PasswordInput(attrs={'class': 'form-
control'}))
```

accounts/models.py:

```
from django.db import models

# Create your models here.
```

Accounts/urls.py:

```
from django.urls import path
from . import views

app_name = "accounts"

urlpatterns=[
    path('login/', views.login_user, name='login'),
    path('logout/', views.logout_user, name='logout'),

]
```

Accounts/views.py:

```
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import render, redirect
from django.contrib.auth.models import User
from .forms import UserRegistrationForm
from django.contrib import messages
from django.http import HttpResponse


def login_user(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            redirect_url = request.GET.get('next', 'home')
            return redirect(redirect_url)
        else:
            messages.error(request, "Username Or Password is incorrect!!",
                           extra_tags='alert alert-warning alert-dismissible fade
show')

    return render(request, 'accounts/login.html')


def logout_user(request):
    logout(request)
    return redirect('home')
```

accounts/templates/accounts/login.html:

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <div class="row center">
        <div class="col-md-6 offset-md-3">
            {% if messages %}
            <div class="messages">
                {% for message in messages %}
                <div {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{ message }}
                    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                {% endfor %}
            </div>
            {% endif %}
            <form action="" method="POST">
                {% csrf_token %}
                <div class="form-group">
                    <label for="username">Username</label>
                    <input type="text" class="form-control" name="username" placeholder="Enter Username">
                </div>
                <div class="form-group">
                    <label for="password">Password</label>
                    <input type="password" class="form-control" name="password" placeholder="Password">
                </div>
                <button type="submit" class="btn btn-primary">Login</button>
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

Another app is called polls:

polls/admin.py:

```
from django.contrib import admin
from .models import Poll, Choice, Vote

admin.site.register(Poll)
admin.site.register(Choice)
admin.site.register(Vote)
```

polls/apps.py:

```
from django.apps import AppConfig


class PollsConfig(AppConfig):
    name = 'polls'
```

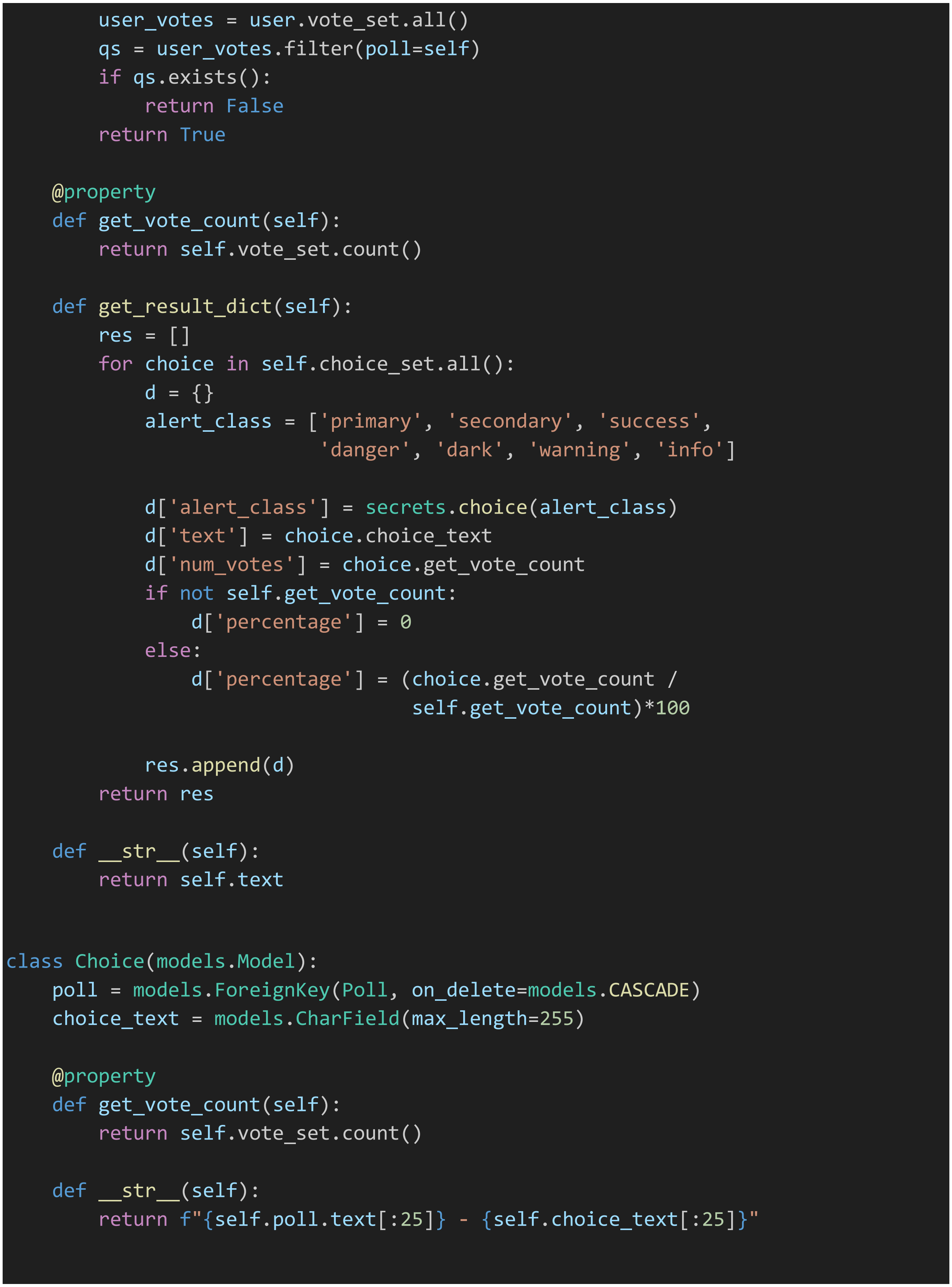
polls/forms.py:

```
from django import forms
from .models import Poll, Choice


class PollAddForm(forms.ModelForm):

    choice1 = forms.CharField(label='Choice 1', max_length=100, min_length=2,
                              widget=forms.TextInput(attrs={'class': 'form-control','style': " background-color: rgba(0, 0, 0, 0.5); color: white;"}))
    choice2 = forms.CharField(label='Choice 2', max_length=100, min_length=2,
                              widget=forms.TextInput(attrs={'class': 'form-control', 'style': " background-color: rgba(0, 0, 0, 0.5); color: white;"}))

    class Meta:
        model = Poll
        fields = ['text', 'choice1', 'choice2']
        widgets = {
            'text': forms.Textarea(attrs={'class': 'form-control', 'rows': 5, 'cols': 20,  'style': " background-color: rgba(0, 0, 0, 0.5); color: white; font-size: 20px;"}),
        }
```

```
class EditPollForm(forms.ModelForm):
    class Meta:
        model = Poll
        fields = ['text', ]
        widgets = {
            'text': forms.Textarea(attrs={'class': 'form-control', 'rows': 2, 'cols': 20, 'style': " background-color: rgba(0, 0, 0, 0.5); color: white; font-size: 20px;"}),
        }


class ChoiceAddForm(forms.ModelForm):
    class Meta:
        model = Choice
        fields = ['choice_text', ]
        widgets = {
            'choice_text': forms.TextInput(attrs={'class': 'form-control', 'style': " background-color: rgba(0, 0, 0, 0.5); color: white;"})
        }
```

polls/models.py:

```
from django.contrib.auth.models import User
from django.db import models
from django.utils import timezone
import secrets


class Poll(models.Model):
    owner = models.ForeignKey(User, on_delete=models.CASCADE)
    text = models.TextField()
    pub_date = models.DateTimeField(default=timezone.now)
    active = models.BooleanField(default=True)

    def user_can_vote(self, user):
        """
        Return False if user already voted
        """
```



```
        user_votes = user.vote_set.all()
        qs = user_votes.filter(poll=self)
        if qs.exists():
            return False
        return True

    @property
    def get_vote_count(self):
        return self.vote_set.count()

    def get_result_dict(self):
        res = []
        for choice in self.choice_set.all():
            d = {}
            alert_class = ['primary', 'secondary', 'success',
                           'danger', 'dark', 'warning', 'info']

            d['alert_class'] = secrets.choice(alert_class)
            d['text'] = choice.choice_text
            d['num_votes'] = choice.get_vote_count
            if not self.get_vote_count:
                d['percentage'] = 0
            else:
                d['percentage'] = (choice.get_vote_count /
                                   self.get_vote_count)*100

            res.append(d)
        return res

    def __str__(self):
        return self.text


class Choice(models.Model):
    poll = models.ForeignKey(Poll, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=255)

    @property
    def get_vote_count(self):
        return self.vote_set.count()

    def __str__(self):
        return f"{self.poll.text[:25]} - {self.choice_text[:25]}"
```

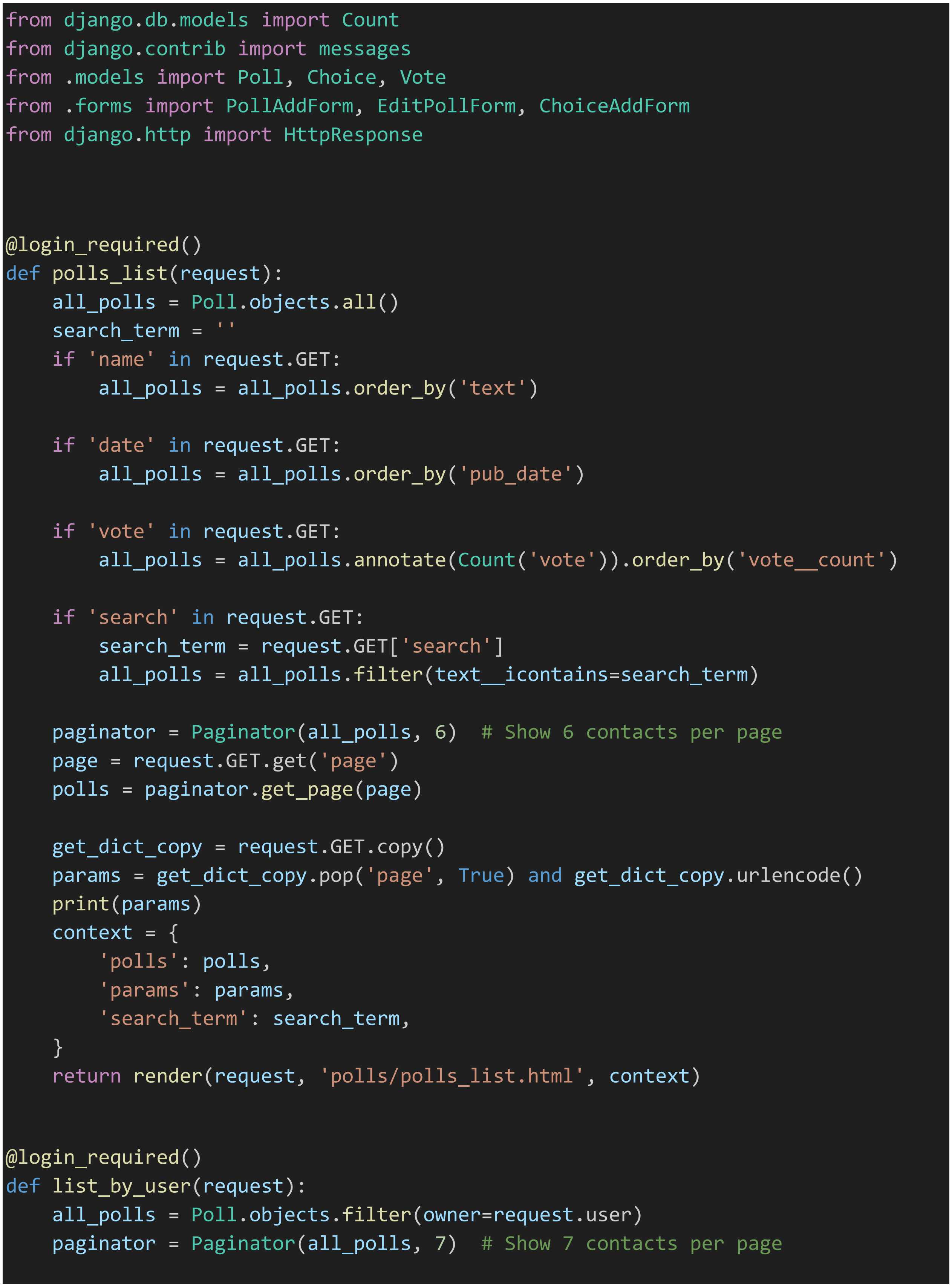
```
class Vote(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    poll = models.ForeignKey(Poll, on_delete=models.CASCADE)
    choice = models.ForeignKey(Choice, on_delete=models.CASCADE)

    def __str__(self):
        return f'{self.poll.text[:15]} - {self.choice.choice_text[:15]} -
{self.user.username}'
```

polls/urls.py:

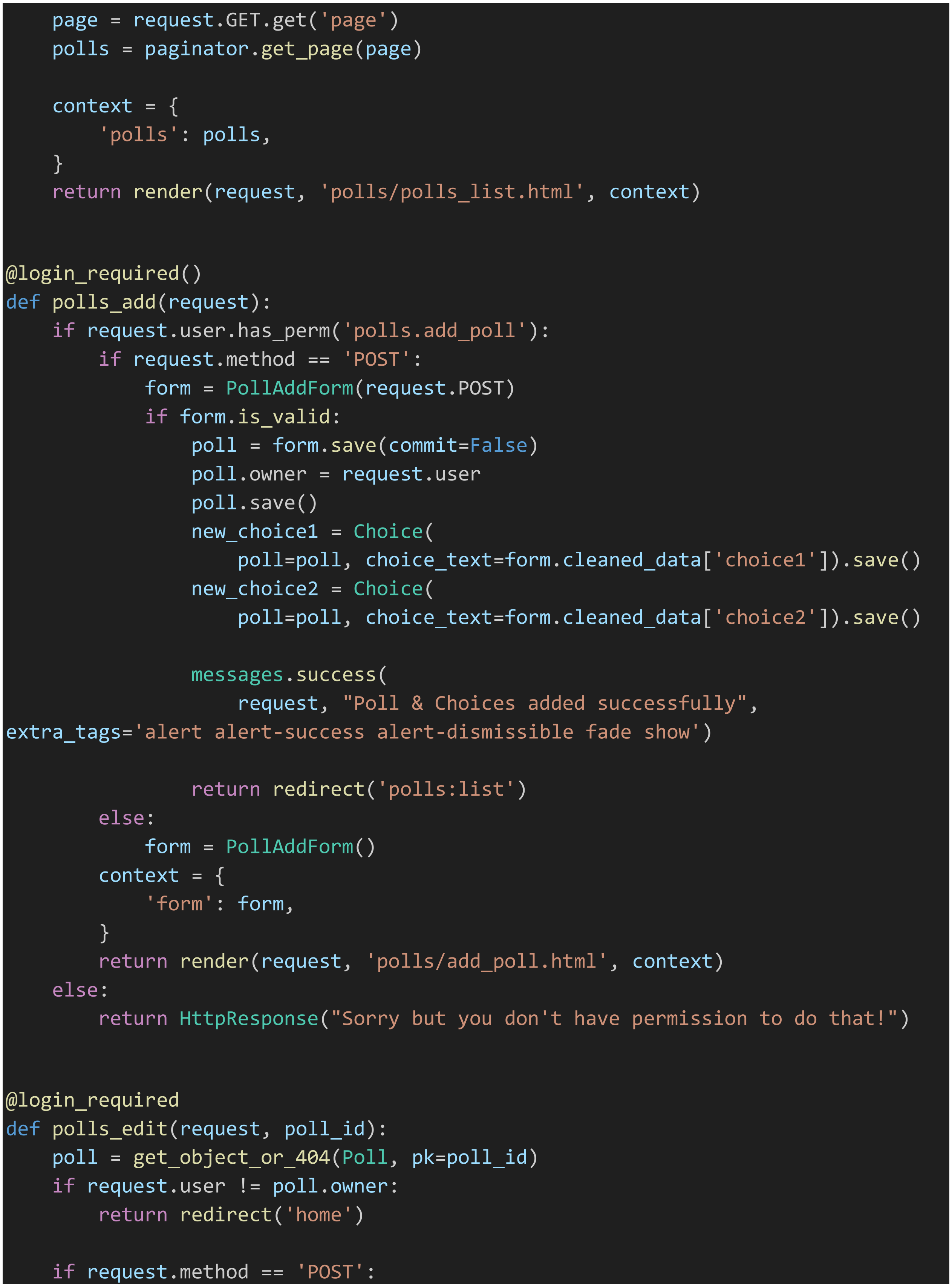
```
from django.urls import path
from . import views

app_name = "polls"

urlpatterns = [
    path('list/', views.polls_list, name='list'),
    path('list/user/', views.list_by_user, name='list_by_user'),
    path('add/', views.polls_add, name='add'),
    path('edit/<int:poll_id>/', views.polls_edit, name='edit'),
    path('delete/<int:poll_id>/', views.polls_delete, name='delete_poll'),
    path('end/<int:poll_id>/', views.endpoll, name='end_poll'),
    path('edit/<int:poll_id>/choice/add/', views.add_choice, name='add_choice'),
    path('edit/choice/<int:choice_id>/', views.choice_edit, name='choice_edit'),
    path('delete/choice/<int:choice_id>/',
         views.choice_delete, name='choice_delete'),
    path('<int:poll_id>/', views.poll_detail, name='detail'),
    path('<int:poll_id>/vote/', views.poll_vote, name='vote'),
]
```

polls/views.py:

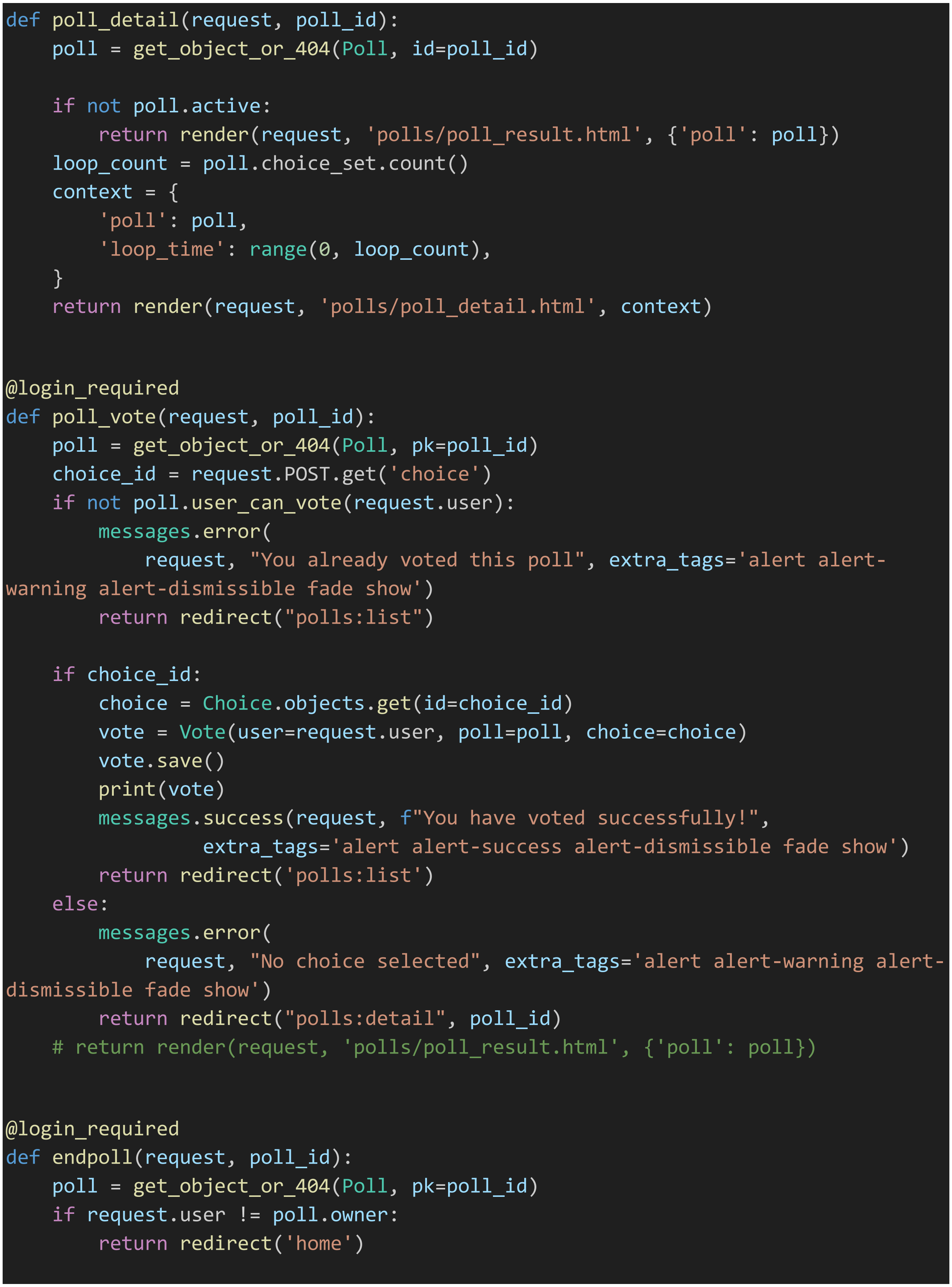
```
from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from django.core.paginator import Paginator
```



```python
from django.db.models import Count
from django.contrib import messages
from .models import Poll, Choice, Vote
from .forms import PollAddForm, EditPollForm, ChoiceAddForm
from django.http import HttpResponse



@login_required()
def polls_list(request):
    all_polls = Poll.objects.all()
    search_term = ''
    if 'name' in request.GET:
        all_polls = all_polls.order_by('text')

    if 'date' in request.GET:
        all_polls = all_polls.order_by('pub_date')

    if 'vote' in request.GET:
        all_polls = all_polls.annotate(Count('vote')).order_by('vote__count')

    if 'search' in request.GET:
        search_term = request.GET['search']
        all_polls = all_polls.filter(text__icontains=search_term)

    paginator = Paginator(all_polls, 6)  # Show 6 contacts per page
    page = request.GET.get('page')
    polls = paginator.get_page(page)

    get_dict_copy = request.GET.copy()
    params = get_dict_copy.pop('page', True) and get_dict_copy.urlencode()
    print(params)
    context = {
        'polls': polls,
        'params': params,
        'search_term': search_term,
    }
    return render(request, 'polls/polls_list.html', context)


@login_required()
def list_by_user(request):
    all_polls = Poll.objects.filter(owner=request.user)
    paginator = Paginator(all_polls, 7)  # Show 7 contacts per page
```



```
    page = request.GET.get('page')
    polls = paginator.get_page(page)

    context = {
        'polls': polls,
    }
    return render(request, 'polls/polls_list.html', context)


@login_required()
def polls_add(request):
    if request.user.has_perm('polls.add_poll'):
        if request.method == 'POST':
            form = PollAddForm(request.POST)
            if form.is_valid:
                poll = form.save(commit=False)
                poll.owner = request.user
                poll.save()
                new_choice1 = Choice(
                    poll=poll, choice_text=form.cleaned_data['choice1']).save()
                new_choice2 = Choice(
                    poll=poll, choice_text=form.cleaned_data['choice2']).save()

                messages.success(
                    request, "Poll & Choices added successfully",
extra_tags='alert alert-success alert-dismissible fade show')

                return redirect('polls:list')
        else:
            form = PollAddForm()
        context = {
            'form': form,
        }
        return render(request, 'polls/add_poll.html', context)
    else:
        return HttpResponse("Sorry but you don't have permission to do that!")


@login_required
def polls_edit(request, poll_id):
    poll = get_object_or_404(Poll, pk=poll_id)
    if request.user != poll.owner:
        return redirect('home')

    if request.method == 'POST':
        form = EditPollForm(request.POST, instance=poll)
        if form.is_valid:
            form.save()
            messages.success(request, "Poll Updated successfully",
                             extra_tags='alert alert-success alert-dismissible fade show')
            return redirect("polls:list")

    else:
        form = EditPollForm(instance=poll)

    return render(request, "polls/poll_edit.html", {'form': form, 'poll': poll})


@login_required
def polls_delete(request, poll_id):
    poll = get_object_or_404(Poll, pk=poll_id)
    if request.user != poll.owner:
        return redirect('home')
    poll.delete()
    messages.success(request, "Poll Deleted successfully",
                     extra_tags='alert alert-success alert-dismissible fade show')
    return redirect("polls:list")


@login_required
def add_choice(request, poll_id):
    poll = get_object_or_404(Poll, pk=poll_id)
    if request.user != poll.owner:
        return redirect('home')

    if request.method == 'POST':
        form = ChoiceAddForm(request.POST)
        if form.is_valid:
            new_choice = form.save(commit=False)
            new_choice.poll = poll
            new_choice.save()
            messages.success(
                request, "Choice added successfully", extra_tags='alert alert-success alert-dismissible fade show')
            return redirect('polls:edit', poll.id)
    else:
        form = ChoiceAddForm()
    context = {
        'form': form,
    }
    return render(request, 'polls/add_choice.html', context)


@login_required
def choice_edit(request, choice_id):
    choice = get_object_or_404(Choice, pk=choice_id)
    poll = get_object_or_404(Poll, pk=choice.poll.id)
    if request.user != poll.owner:
        return redirect('home')

    if request.method == 'POST':
        form = ChoiceAddForm(request.POST, instance=choice)
        if form.is_valid:
            new_choice = form.save(commit=False)
            new_choice.poll = poll
            new_choice.save()
            messages.success(
                request, "Choice Updated successfully", extra_tags='alert alert-success alert-dismissible fade show')
            return redirect('polls:edit', poll.id)
    else:
        form = ChoiceAddForm(instance=choice)
    context = {
        'form': form,
        'edit_choice': True,
        'choice': choice,
    }
    return render(request, 'polls/add_choice.html', context)


@login_required
def choice_delete(request, choice_id):
    choice = get_object_or_404(Choice, pk=choice_id)
    poll = get_object_or_404(Poll, pk=choice.poll.id)
    if request.user != poll.owner:
        return redirect('home')
    choice.delete()
    messages.success(
        request, "Choice Deleted successfully", extra_tags='alert alert-success alert-dismissible fade show')
    return redirect('polls:edit', poll.id)
```



```
def poll_detail(request, poll_id):
    poll = get_object_or_404(Poll, id=poll_id)

    if not poll.active:
        return render(request, 'polls/poll_result.html', {'poll': poll})
    loop_count = poll.choice_set.count()
    context = {
        'poll': poll,
        'loop_time': range(0, loop_count),
    }
    return render(request, 'polls/poll_detail.html', context)


@login_required
def poll_vote(request, poll_id):
    poll = get_object_or_404(Poll, pk=poll_id)
    choice_id = request.POST.get('choice')
    if not poll.user_can_vote(request.user):
        messages.error(
            request, "You already voted this poll", extra_tags='alert alert-
warning alert-dismissible fade show')
        return redirect("polls:list")

    if choice_id:
        choice = Choice.objects.get(id=choice_id)
        vote = Vote(user=request.user, poll=poll, choice=choice)
        vote.save()
        print(vote)
        messages.success(request, f"You have voted successfully!",
                         extra_tags='alert alert-success alert-dismissible fade show')
        return redirect('polls:list')
    else:
        messages.error(
            request, "No choice selected", extra_tags='alert alert-warning alert-
dismissible fade show')
        return redirect("polls:detail", poll_id)
    # return render(request, 'polls/poll_result.html', {'poll': poll})


@login_required
def endpoll(request, poll_id):
    poll = get_object_or_404(Poll, pk=poll_id)
    if request.user != poll.owner:
        return redirect('home')
    if poll.active is True:
        poll.active = False
        poll.save()
        return render(request, 'polls/poll_result.html', {'poll': poll})
    else:
        return render(request, 'polls/poll_result.html', {'poll': poll})
```

polls/templates/polls:

polls/templates/polls /add_choice.html:

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <div class="row center">
        <div class="col-md-6 offset-md-3">
            {% if edit_choice %}
            <h2>Update choice</h2>
            {% else %}
            <h2>Add new choice</h2>
            {% endif %}
            {% if messages %}
            <ul class="messages">
                {% for message in messages %}
                <li {% if message.tags %} class="{{ message.tags }}" {% endif
%}>{{ message }}</li>
                {% endfor %}
            </ul>
            {% endif %}
            <form action="" method="POST">
                {% csrf_token %}
                {% for field in form %}
                <div class="form-group">
                    {{ field.errors }}
                    {{ field.label_tag }}
                    {{ field }}
                </div>
                {% endfor %}
                {% if edit_choice %}
                <button type="submit" class="btn btn-primary">Update</button>
                <a class="btn btn-danger" href="{% url 'polls:choice_delete' 
choice.id %}" role="button" onclick="return confirm('Are you sure you want to 
delete this?')">Delete</a>
                {% else %}
                <button type="submit" class="btn btn-primary">Add</button>
                {% endif %}
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

polls/templates/polls /add_poll.html:

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <div class="row center">
        <div class="col-md-6 offset-md-3">
            <h2>Create new poll</h2>
            {% if messages %}
            <ul class="messages">
                {% for message in messages %}
                <li {% if message.tags %} class="{{ message.tags }}" {% endif 
%}>{{ message }}</li>
                {% endfor %}
            </ul>
            {% endif %}
            <form action="" method="POST">
                {% csrf_token %}
                {% for field in form %}
                <div class="form-group">
                    {{ field.errors }}
                    {{ field.label_tag }}
                    {{ field }}
                </div>
                {% endfor %}
                <button type="submit" class="btn btn-primary">Add Poll</button>
                <a class="btn btn-warning" href="{% url 'polls:list' %}" 
role="button">Back</a>
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

polls/templates/polls /end_poll.html:

```
{% extends 'base.html' %}


{% block content %}
<div class="container">
    {% if messages %}
    <div class="messages">
        {% for message in messages %}
        <div {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{
message }}
            <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>
        {% endfor %}
    </div>
    {% else %}
    <div class="row">
        <div class="col-md-8 offset-sm-2">
            <h3 class="mt-3 mb-3 text-center">Result for: {{ poll.text }}</h3>
            <!-- progress bar -->
    <div class="progress mt-3">

        {% for choice in poll.get_result_dict %}
        <div class="progress-bar bg-{{ choice.alert_class }}" role="progressbar"
style="width: {{ choice.percentage }}%;" aria-valuenow="30" aria-valuemin="0"
            aria-valuemax="100"><b>{{ choice.text }}-{{
choice.percentage|floatformat }}%</b></div>
        {% endfor %}

    </div>
            <ul class="list-group">
                {% for choice in poll.choice_set.all %}
                <li class="list-group-item d-flex justify-content-between align-
items-center">
                    {{ choice.choice_text }}
                    <span class="badge badge-primary badge-pill">{{
choice.get_vote_count }}</span>
```

```
                </li>
                {% endfor %}
            </ul>
            {% endif %}
            <a class="btn btn-primary mt-3" href="{% url 'polls:list' %}" role="button">Back To Polls</a>
        </div>

    </div>
</div>
{% endblock content %}
```

polls/templates/polls /poll_detail.html:

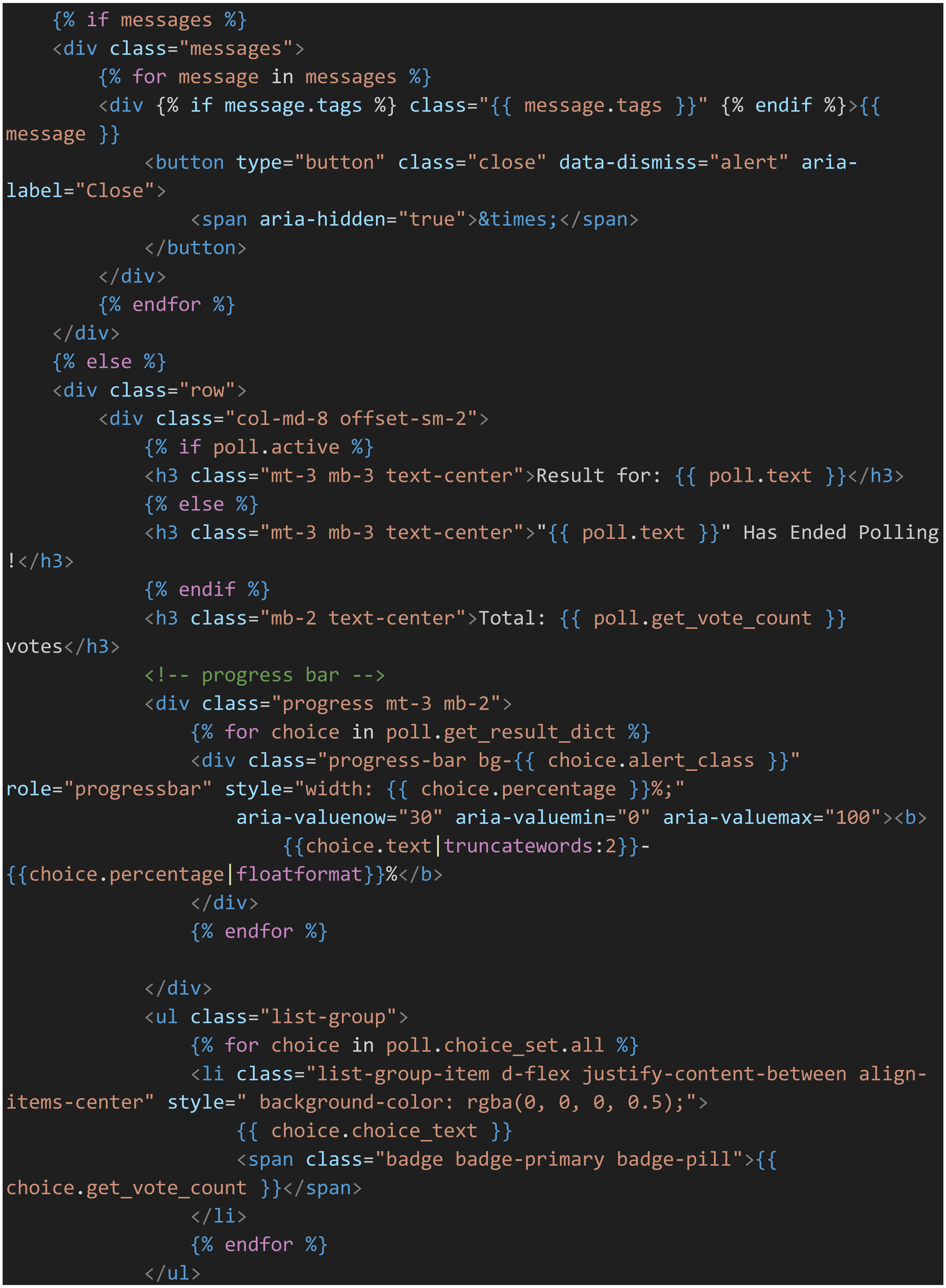
```
{% extends 'base.html' %}

{% block content %}
<div class="container d-flex align-items-center justify-content-center" style="min-height: 100vh; font-family: 'Roboto', sans-serif;">
    <div style="text-align: left; width: 300px;"> <!-- Adjust the width as needed -->
        <h1 class="text-center mt-3 mb-3">VOTE</h1>
        {% if messages %}
        <div class="messages">
            {% for message in messages %}
            <div {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{ message }}
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            {% endfor %}
        </div>
        {% endif %}
        <hr>
        <h2 class="mt-3 mb-3">{{ poll }}</h2>
        <form action="{% url 'polls:vote' poll.id %}" method="POST" >
            {% csrf_token %}
            {% for choice in poll.choice_set.all %}
            <div class="form-check list-group-item" style=" background-color:
rgba(0, 0, 0, 0.4);">
                <input type="radio" name="choice" id="choice{{ forloop.counter
}}" value="{{ choice.id }}" class="form-check-input">
                <label for="choice{{ forloop.counter }}" class="form-check-
label">{{ choice.choice_text }}</label>
            </div>
            {% endfor %}
            <input type="submit" value="Vote" class="btn btn-primary mt-3">
            <a class="btn btn-warning mt-3" href="{% url 'polls:list' %}"
role="button">Cancel</a>

        </form>
    </div>
</div>

{% endblock content %}
```

polls/templates/polls /poll_edit.html:

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <div class="row center">
        <div class="col-md-6 offset-md-3">
            <h2>Edit poll</h2>
            {% if messages %}
            {% if messages %}
            <div class="messages">
                {% for message in messages %}
                <div {% if message.tags %} class="{{ message.tags }}" {% endif
%}>{{ message }}
                    <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                {% endfor %}
            </div>
            {% endif %}
            {% endif %}
            <form action="" method="POST">
                {% csrf_token %}
                {% for field in form %}
                <div class="form-group" >
                    {{ field.errors }}
                    {{ field.label_tag }}
                    {{ field }}
                </div>
                {% endfor %}
                <button type="submit" class="btn btn-primary">Update</button>
                <a class="btn btn-danger" href="{% url 'polls:delete_poll' poll.id %}" role="button" onclick="return confirm('Are you sure?')">Delete</a>
                <a class="btn btn-warning" href="{% url 'polls:add_choice' poll.id %}" role="button">Add Choice</a>
            </form>

            <div class="choices">
                <h2 class="text-center mt-3">Choices</h2>
                <hr>
                <ul class="list-group" >
                    {% for choice in poll.choice_set.all %}
                    <li class="list-group-item" style=" background-color: rgba(0, 0, 0, 0.5);"><a href="{% url 'polls:choice_edit' choice.id %}"><i class="fas fa-pencil-alt"></i></a>&nbsp
                        {{ choice.choice_text }}</li>
                    {% endfor %}
                </ul>
            </div>

        </div>
    </div>
</div>
{% endblock %}
```

polls/templates/polls /poll_result.html:

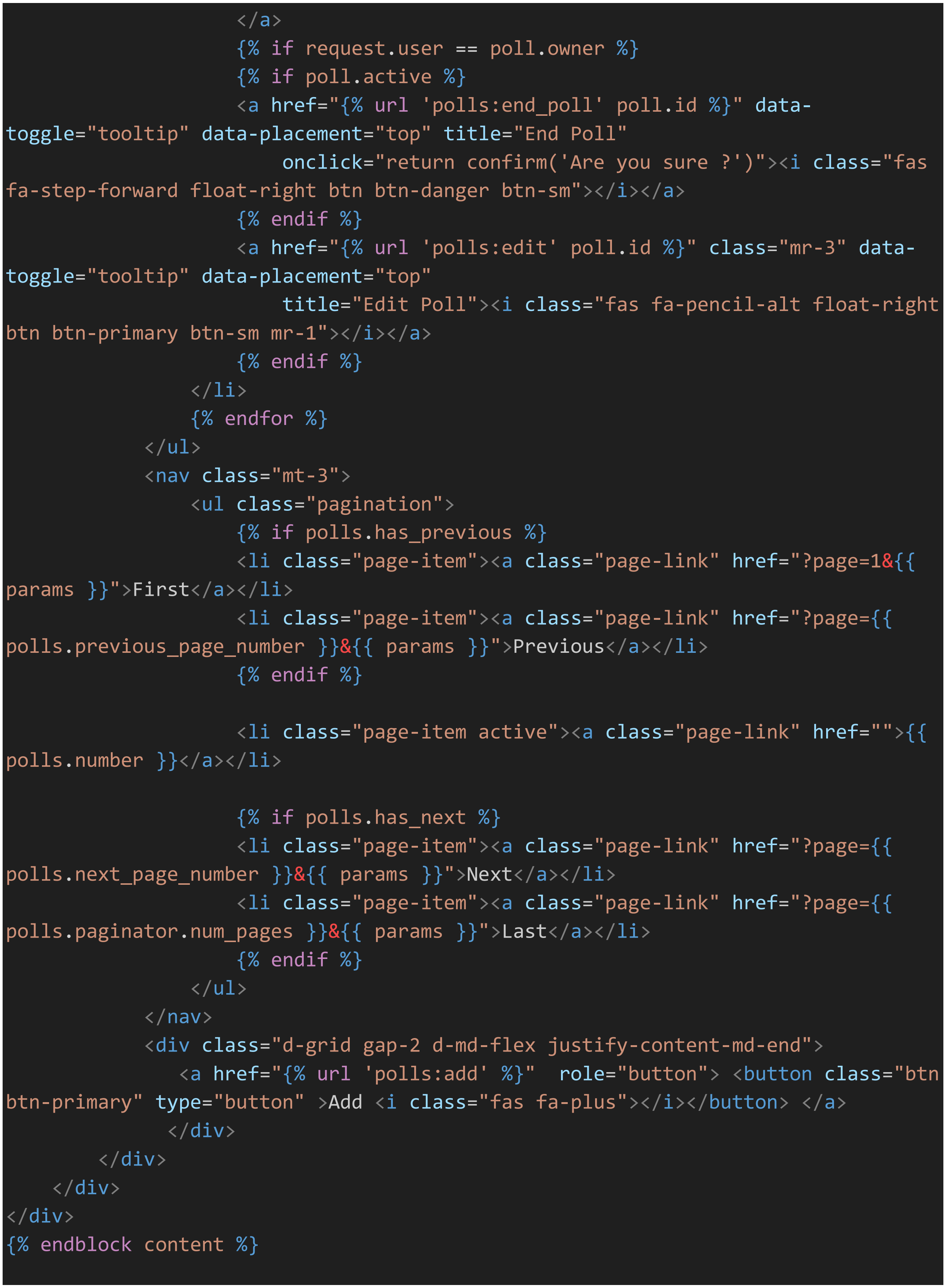
```
{% extends 'base.html' %}

{% block content %}
<div class="container">
```



```
    {% if messages %}
    <div class="messages">
        {% for message in messages %}
        <div {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{
message }}
            <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>
        {% endfor %}
    </div>
    {% else %}
    <div class="row">
        <div class="col-md-8 offset-sm-2">
            {% if poll.active %}
            <h3 class="mt-3 mb-3 text-center">Result for: {{ poll.text }}</h3>
            {% else %}
            <h3 class="mt-3 mb-3 text-center">"{{ poll.text }}" Has Ended Polling
!</h3>
            {% endif %}
            <h3 class="mb-2 text-center">Total: {{ poll.get_vote_count }}
votes</h3>
            <!-- progress bar -->
            <div class="progress mt-3 mb-2">
                {% for choice in poll.get_result_dict %}
                <div class="progress-bar bg-{{ choice.alert_class }}"
role="progressbar" style="width: {{ choice.percentage }}%;"
                    aria-valuenow="30" aria-valuemin="0" aria-valuemax="100"><b>
                        {{choice.text|truncatewords:2}}-
{{choice.percentage|floatformat}}%</b>
                </div>
                {% endfor %}

            </div>
            <ul class="list-group">
                {% for choice in poll.choice_set.all %}
                <li class="list-group-item d-flex justify-content-between align-
items-center" style=" background-color: rgba(0, 0, 0, 0.5);">
                    {{ choice.choice_text }}
                    <span class="badge badge-primary badge-pill">{{
choice.get_vote_count }}</span>
                </li>
                {% endfor %}
            </ul>
            {% endif %}
            <a class="btn btn-primary mt-3" href="{% url 'polls:list' %}" role="button">Back To Polls</a>
        </div>

    </div>
</div>
{% endblock content %}
```

polls/templates/polls /polls_list.html:

```
{% extends 'base.html' %}

{% block content %}
<div class="container">
    <div class="row">
        <div class="col-md-8 offset-sm-2">
            <div class="d-flex justify-content-between align-items-center">
                <h1 class="text-center mt-3 mb-3">Ballot </h1>
                {% if messages %}
                <div class="messages">
                    {% for message in messages %}
                    <div {% if message.tags %} class="{{ message.tags }}" {% endif %}>{{ message }}
                        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                            <span aria-hidden="true">&times;</span>
                        </button>
                    </div>
                    {% endfor %}
                </div>
                {% endif %}

            </div>

            <ul class="list-group">
                {% for poll in polls %}
                <li class="list-group-item" style=" background-color: rgba(0, 0, 0, 0.5);"><a href="{% url 'polls:detail' poll.id %}" style="color: white; font-size: 20px;">{{ poll.text|truncatewords:5 }}
                        {% if not poll.active%}
                        <i class="fas fa-check-circle ml-2"></i>
                        {% endif %}
```



```
                    </a>
                    {% if request.user == poll.owner %}
                    {% if poll.active %}
                    <a href="{% url 'polls:end_poll' poll.id %}" data-
toggle="tooltip" data-placement="top" title="End Poll"
                        onclick="return confirm('Are you sure ?')"><i class="fas
fa-step-forward float-right btn btn-danger btn-sm"></i></a>
                    {% endif %}
                    <a href="{% url 'polls:edit' poll.id %}" class="mr-3" data-
toggle="tooltip" data-placement="top"
                        title="Edit Poll"><i class="fas fa-pencil-alt float-right
btn btn-primary btn-sm mr-1"></i></a>
                    {% endif %}
                </li>
                {% endfor %}
            </ul>
            <nav class="mt-3">
                <ul class="pagination">
                    {% if polls.has_previous %}
                    <li class="page-item"><a class="page-link" href="?page=1&{{
params }}">First</a></li>
                    <li class="page-item"><a class="page-link" href="?page={{
polls.previous_page_number }}&{{ params }}">Previous</a></li>
                    {% endif %}

                    <li class="page-item active"><a class="page-link" href="">{{
polls.number }}</a></li>

                    {% if polls.has_next %}
                    <li class="page-item"><a class="page-link" href="?page={{
polls.next_page_number }}&{{ params }}">Next</a></li>
                    <li class="page-item"><a class="page-link" href="?page={{
polls.paginator.num_pages }}&{{ params }}">Last</a></li>
                    {% endif %}
                </ul>
            </nav>
            <div class="d-grid gap-2 d-md-flex justify-content-md-end">
               <a href="{% url 'polls:add' %}"  role="button"> <button class="btn
btn-primary" type="button" >Add <i class="fas fa-plus"></i></button> </a>
              </div>
        </div>
    </div>
</div>
{% endblock content %}
```

CSS:

```css
html, body {
    height:100%;
    padding: 0;
    margin: 0;
    color: white;
  }
html{
    background-image: url('../img/background.jpg');
    background-size: cover;
}
body {
    background: rgba(0, 0, 0, 0.466);
}
#home-content{
    text-align: center;
    padding-top: 20%;
}
```

Then in the base is a templates folder with the following

templates/include/navbar.html:

```html
<nav class="navbar navbar-expand-sm navbar-light bg-light">
    <a class="navbar-brand" href=""><i class="fas fa-person-booth"></i></a>
    <button class="navbar-toggler d-lg-none" type="button" data-toggle="collapse" data-target="#collapsibleNavId"
        aria-controls="collapsibleNavId" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavId">
        <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
            {% comment %} <li class="nav-item active">
                <a class="nav-link" href="{% url 'home' %}">Home <span class="sr-only">(current)</span></a>
            </li> {% endcomment %}
            <li class="nav-item">
                <a class="nav-link" href="{% url 'polls:list' %}">Polls</a>
            </li>
        </ul>
        <div class="navbar-nav ml-auto">
            {% if request.user.is_authenticated %}
            <a class="nav-link" href="{% url 'polls:list_by_user' %}">My
Polls</a>
            <a class="nav-link" href="{% url 'accounts:logout' %}">Logout</a>
            {% else %}
            <a class="nav-link" href="{% url 'accounts:login' %}">Login</a>
            {% endif %}
        </div>
    </div>
</nav>
```

Then in templates located in the base again lies two html files:

base.html:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Poll Me | Polls List</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
        crossorigin="anonymous">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-
UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/"
        crossorigin="anonymous">
        <link rel="stylesheet" href="{% static 'css/home_style.css' %}">
    {% block custom_css %}{% endblock custom_css %}
```

```
</head>

<body>
    {% include 'includes/navbar.html' %}

    {% block content %}

    {% endblock content %}


    <!-- All javascript files goes under here    -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
        crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
        crossorigin="anonymous"></script>

</body>

</html>
```

Then also Home.html:

templates/ Home.html:

```
{% extends 'base.html' %}
{% load static %}
{% block custom_css %}
<link rel="stylesheet" href="{% static 'css/home_style.css' %}">
{% endblock custom_css %}


{% block content %}
<div class="container">
    <div class="row">
```

```
        <div class="col">
            <div id="home-content">
                <h1>PollMe - Get Started!</h1>
                <h3>Your Voice Matters!</h3>
                <hr>
                <a class="btn btn-light btn-lg" href="{% url 'polls:list' %}"
role="button">Get Started !</a>
            </div>
        </div>
    </div>
</div>
{% endblock content %}
```

```
pollme/

|

├── manage.py

|

├── pollme/

|   ├── __init__.py

|   ├── settings.py

|   ├── urls.py

|   └── wsgi.py

|

├── polls/
```

```
│   ├── migrations/
│   │   └── ...
│   ├── static/
│   │   ├── css/
│   │   │   └── home_style.css
│   │   └── img/
│   │       └── background.jpg
│   ├── templates/
│   │   ├── includes/
│   │   │   └── navbar.html
│   │   ├── add_choice.html
│   │   ├── add_poll.html
│   │   ├── end_poll.html
│   │   ├── poll_detail.html
│   │   ├── poll_edit.html
│   │   ├── poll_result.html
│   │   ├── polls_list.html
│   │   └── ...
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── models.py
│   ├── tests.py
│   ├── urls.py
│   └── views.py
│
├── accounts/
│   ├── migrations/
│   │   └── ...
```

```
|   ├── templates/
|   |   └── ...
|   ├── __init__.py
|   ├── admin.py
|   ├── apps.py
|   ├── forms.py
|   ├── models.py
|   ├── tests.py
|   ├── urls.py
|   └── views.py
|
└── templates/
    ├── base.html
    └── home.html
```