**ვირტუალური გარემო და აქტივიზაცია:**

ლინუქსი და მაკი:

python3 -m venv env

source env/bin/activate

ვინდოუსი:

py -m venv env

.\env\Scripts\activate

python manage.py runserver

დეატივიცაზია:

deactivate.

ჯანგოს დაყენება:

python -m pip install Django

ჯანგო:

django-admin startproject mysite (ბოლო პროექტის შერჩეული სახელია)

python manage.py runserver (სერვერის გაშვება, უნდა ვიდგეთ სადავ manage.py ფაილია)

python manage.py runserver 8080 (თუ გვინდა სხვა პორტზე გავუშვათ)

python manage.py runserver 0:8000 (to listen on all available public Ips)

python -m django --version (ჯანგოს ვერსიის გაგება)

python manage.py startapp appname (ბოლო აპის სახელია რომელსაც ვქმნით)

<http://127.0.0.1:8000/> (ინტერნეტში გახსნა ჯანგოს პორტის)

python manage.py makemigrations (მიგრაციის მომზადება)

python manage.py makemigrations polls (კონკრეტული polls -ის მიგრაცია)

python manage.py sqlmigrate polls 0001 (შევხედოთ როგორ წერია SQL-ში)

python manage.py migrate (მიგრაცია)

ჯანგოს შელში შესვლა კომანდ ლაინიდან

python manage.py shell

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

INSTALLED\_APPS = [

'polls.apps.PollsConfig',

Question.objects.all()

Question.objects.filter(question\_text\_\_startswith='What')

**>>> from** **django.utils** **import** timezone

**>>>** current\_year = timezone.now().year

**>>>** Question.objects.get(pub\_date\_\_year=current\_year)

**>>>** Question.objects.get(id=2)

q = Question.objects.get(pk=1)

q.choice\_set.all()

q.choice\_set.create(choice\_text='Not much', votes=0)

q.choice\_set.count()

**>>>** c = q.choice\_set.filter(choice\_text\_\_startswith='Just hacking')

**>>>** c.delete()

polls/admin.py[**¶**](https://docs.djangoproject.com/en/3.2/intro/tutorial02/#id6)

**from** **django.contrib** **import** admin

**from** **.models** **import** Question

admin.site.register(Question)

ჯანგოს ადმინი:

python manage.py createsuperuser

polls/admin.py ფაილში

from django.contrib import admin

from .models import Question, Choice

admin.site.register(Question)

admin.site.register(Choice)

polls/urls.py ფაილი

polls/urls.py¶

from django.urls import path

from . import views

app\_name = 'polls'

urlpatterns = [

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

path('', views.IndexView.as\_view(), name='index'), (ან ეს ვარიანტი)

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'),

]

polls/views.py

from django.http import HttpResponseRedirect

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

from django.urls import reverse

from django.views import generic

from .models import Choice, Question

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'

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,)))

ტესტირება აპის: