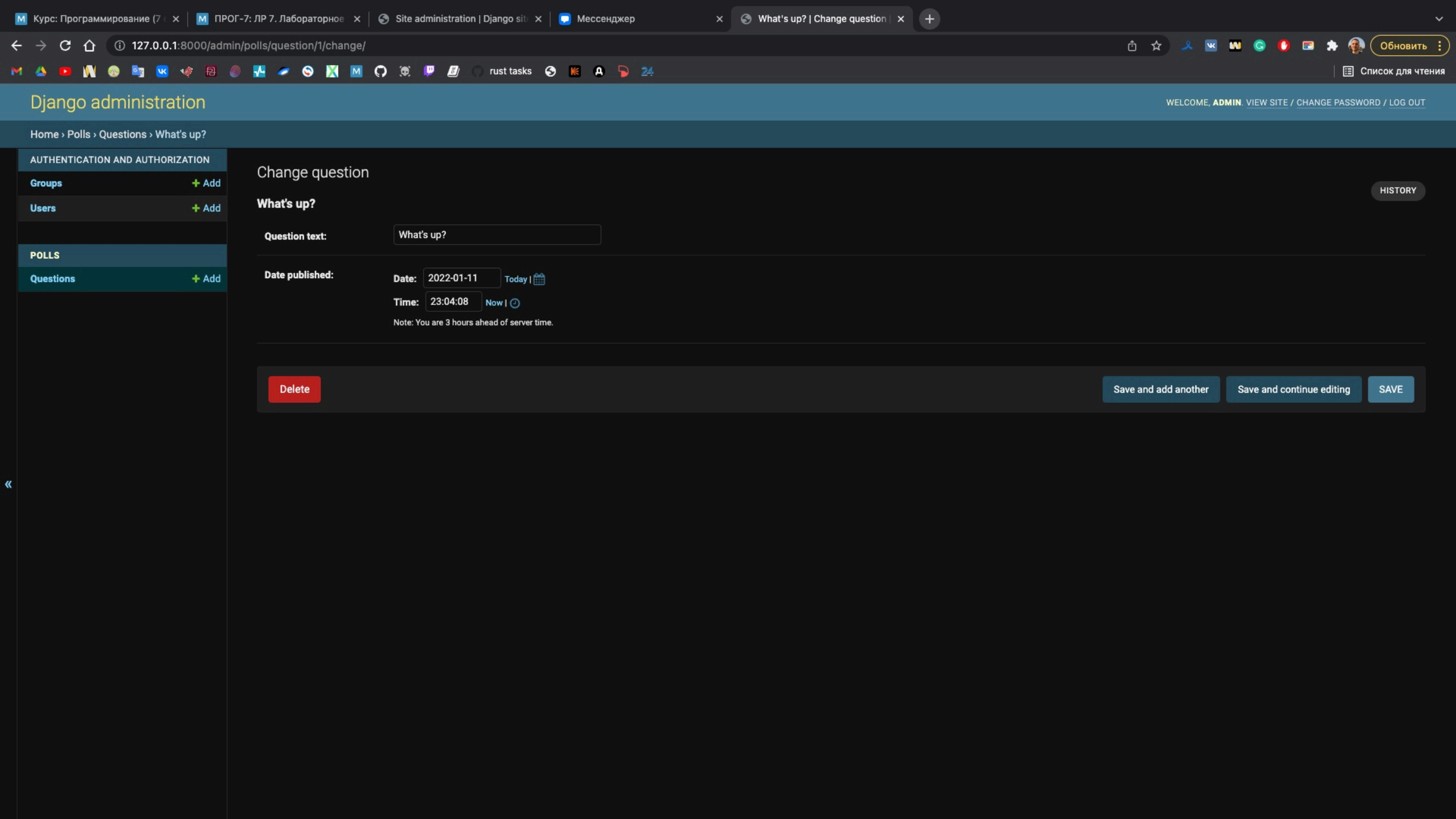


Hello, world. You're at the polls index.



# Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Polls > Questions > What's up?

## AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

## POLLS

Questions [+ Add](#)

## Change question

HISTORY

### What's up?

Question text:

What's up?

Date published:

Date: 2022-01-11

Today |

Time: 23:04:08

Now |

Note: You are 3 hours ahead of server time.

Delete

Save and add another

Save and continue editing

SAVE







EXPLORER

LR7

sitelr

polls

\_\_pycache\_\_

migrations

templates/polls

detail.html

index.html

results.html

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

sitelr

\_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

views.py 4

results.html

detail.html

index.htr

tests.py 1

models.py 3

admin.py 1

settings.py

urls.

sitelr > polls > views.py > IndexView

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

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:

PROBLEMS 12

OUTPUT

TERMINAL

DEBUG CONSOLE

zsh - sitelr

[11/Jan/2022 23:17:29] "GET /static/admin/js/calendar.js HTTP/1.1" 200 8466

[11/Jan/2022 23:17:29] "GET /static/admin/css/forms.css HTTP/1.1" 200 8804

[11/Jan/2022 23:17:29] "GET /static/admin/js/prepopulate\_init.js HTTP/1.1" 200 492

[11/Jan/2022 23:17:29] "GET /static/admin/js/admin/DateTimeShortcuts.js HTTP/1.1" 200 19634

[11/Jan/2022 23:17:29] "GET /admin/jsi18n/ HTTP/1.1" 200 3195

[11/Jan/2022 23:17:29] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11097

[11/Jan/2022 23:17:29] "GET /static/admin/js/change\_form.js HTTP/1.1" 200 606

[11/Jan/2022 23:17:29] "GET /static/admin/img/icon-calendar.svg HTTP/1.1" 200 1086

[11/Jan/2022 23:17:29] "GET /static/admin/img/icon-clock.svg HTTP/1.1" 200 677

^C

/Users/denisnyux/Documents/School/Programming/4kurs/lr7/sitelr ~

Python 3.9.7 64-bit

0

12

Ln 10, Col 24

Spaces: 4

UTF-8

LF

Python



EXPLORER

LR7

sitelr

polls

> \_\_pycache\_\_

> migrations

templates/polls

<> detail.html

<> index.html

<> results.html

\_\_init\_\_.py

admin.py 1

apps.py 1

models.py 2

tests.py 2

urls.py 1

views.py 4

sitelr

> \_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py 2

wsgi.py

db.sqlite3

manage.py

nl

detail.html

index.html

tests.py 2

models.py 2

admin.py 1

apps.py 1

settings.py

urls.py .../polls 1

sitelr > polls > tests.py > QuestionModelTests > test\_was\_published\_recently\_with\_future\_question

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

class QuestionModelTests(TestCase):

def test\_was\_published\_recently\_with\_future\_question(self):

"""

was\_published\_recently() returns False for questions whose pub\_date

is in the future.

"""

time = timezone.now() + datetime.timedelta(days=30)

future\_question = Question(pub\_date=time)

self.assertIs(future\_question.was\_published\_recently(), False)

def test\_was\_published\_recently\_with\_old\_question(self):

"""

was\_published\_recently() returns False for questions whose pub\_date

is older than 1 day.

"""

time = timezone.now() - datetime.timedelta(days=1, seconds=1)

old\_question = Question(pub\_date=time)

self.assertIs(old\_question.was\_published\_recently(), False)

PROBLEMS 13

OUTPUT

TERMINAL

DEBUG CONSOLE

zsh - sitelr

+

^

×

AssertionError: True is not False

-----

Ran 1 test in 0.001s

FAILED (failures=1)

Destroying test database for alias 'default'...

/Users/denisnyux/Documents/School/Programming/4kurs/lr7/sitelr ~ python3 manage.py test polls

Creating test database for alias 'default'...

System check identified no issues (0 silenced).

.

-----

Ran 1 test in 0.001s

OK

Destroying test database for alias 'default'...

/Users/denisnyux/Documents/School/Programming/4kurs/lr7/sitelr ~ python3 manage.py test polls

Creating test database for alias 'default'...

System check identified no issues (0 silenced).

...

-----

Ran 3 tests in 0.001s

OK

Destroying test database for alias 'default'...

/Users/denisnyux/Documents/School/Programming/4kurs/lr7/sitelr ~

> OUTLINE

Python 3.9.7 64-bit

⊗ 0 ▲ 13

Ln 12, Col 12

Spaces: 4

UTF-8

LF

Python



EXPLORER

LR7

sitelr

polls

\_\_pycache\_\_

migrations

templates/polls

detail.html

index.html

results.html

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

sitelr

\_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

OUTLINE

views.py 5

results.html

detail.html

index.html

tests.py 3

models.py 2

admin.py 1

apps.py 1

settin

sitelr > polls > tests.py > QuestionDetailViewTests > test\_future\_question

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

import datetime

from django.test import TestCase

from django.utils import timezone

from django.urls import reverse

from .models import Question

class QuestionModelTests(TestCase):

def test\_was\_published\_recently\_with\_future\_question(self):

"""

was\_published\_recently() returns False for questions whose pub\_date

is in the future.

"""

time = timezone.now() + datetime.timedelta(days=30)

future\_question = Question(pub\_date=time)

self.assertIs(future\_question.was\_published\_recently(), False)

def test\_was\_published\_recently\_with\_old\_question(self):

"""

was\_published\_recently() returns False for questions whose pub\_date

is older than 1 day.

"""

time = timezone.now() - datetime.timedelta(days=1, seconds=1)

old\_question = Question(pub\_date=time)

self.assertIs(old\_question.was\_published\_recently(), False)

def test\_was\_published\_recently\_with\_recent\_question(self):

"""

was\_published\_recently() returns True for questions whose pub\_date

is within the last day.

"""

time = timezone.now() - datetime.timedelta(hours=23, minutes=59, seconds=59)

recent\_question = Question(pub\_date=time)

self.assertIs(recent\_question.was\_published\_recently(), True)

Python 3.9.7 64-bit

0

15

Ln 113, Col 60

Spaces: 4

UTF-8

LF

Python



