

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ САНКТ-ПЕТЕРБУРГСКИЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»

Направление подготовки 09.03.03 «Мобильные и сетевые технологии»

## **ОТЧЕТ**

Тема задания: Реализация простого сайта средствами DJANGO

**Выполнил:**

Студент: Барабанов Д.К., К33402

**Проверил:**

Преподаватель: Говоров А.И.

Санкт-Петербург

2020

**Цель:** овладеть практическими навыками и умениями реализации webсервисов средствами Django 2.2.

1) Добавим модели пользователя Teacher и Student и сделаем страницы регистрации и входа

models.py

```
homework > models.py > Assignment > _str_
1 from django.db import models
2 from django.conf import settings
3 from django.contrib.auth.models import AbstractUser
4
5 subjects = [
6     ('Mathematics', 'Mathematics'),
7     ('Physics', 'Physics'),
8     ('Biology', 'Biology'),
9     ('Informatics', 'Informatics')
10 ]
11
12 group_numbers = [
13     ('1', '1'),
14     ('2', '2'),
15     ('3', '3'),
16     ('4', '4'),
17 ]
18
19 roles = [
20     ('teacher', 'учитель'),
21     ('student', 'студент'),
22 ]
23
24
25 class CustomUser(AbstractUser):
26     role = models.CharField(max_length=10, choices=roles, null=True)
27     group_number = models.CharField(max_length=6, choices=group_numbers, null=True)
28     subject = models.CharField(max_length=50, choices=subjects, null=True)
29     name = models.CharField(max_length=100, null=True)
30     def __str__(self):
31         return '{}'.format(self.name)
32
33
34 class Teacher(models.Model):
35     name = models.CharField(max_length=100, null=True)
36     subject = models.CharField(max_length=50, choices=subjects, null=True)
37     def __str__(self):
38         return '{}'.format(self.name)
39
40
41 class Student(models.Model):
42     name = models.CharField(max_length=100, null=True)
43     group_number = models.CharField(max_length=6, choices=group_numbers, null=True)
44     def __str__(self):
45         return '{}'.format(self.name)
46
47
48 class Homework(models.Model):
49     group_number = models.CharField(max_length=6, choices=group_numbers, null=True)
50     subject = models.CharField(max_length=50, choices=subjects)
51     teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
52     deadline = models.DateField()
53     description = models.TextField(max_length=10000)
54
55     def __str__(self):
56         return '{} {}'.format(self.group_number, self.subject)
57
58
59 class Assignment(models.Model):
60     student = models.ForeignKey(Student, on_delete=models.CASCADE)
61     homework = models.ForeignKey(Homework, on_delete=models.CASCADE)
62     solution = models.TextField(max_length=10000)
63
64     def __str__(self):
65         return '{} - {}'.format(self.student, self.homework.subject)
66
67
68 class Assessment(models.Model):
69     student = models.ForeignKey(Student, on_delete=models.CASCADE)
70     homework = models.ForeignKey(Homework, on_delete=models.CASCADE)
71     grade = models.CharField(max_length=2)
72
73     def __str__(self):
74         return '{} - {}'.format(self.student, self.homework.subject)
```

## urls.py

```
1 from django.urls import path
2 from .views import *
3
4 urlpatterns = [
5     path('', LoginFormView.as_view()),
6     path('login/', LoginFormView.as_view()),
7     path('register/teacher', TeacherRegisterFormView.as_view()),
8     path('register/student', StudentRegisterFormView.as_view()),
9     path('profile/', LoginPass),
10    path('logout/', LogoutView.as_view()),
11
12    path('homework/add/', AddHomework.as_view()),
13    path('homework/<int:pk>/', HomeworkDetail.as_view(), name='teacher_detail'),
14    path('homework/<int:pk>/delete/', DeleteHomework.as_view()),
15    path('homeworks/', Homeworks.as_view()),
16    path('homework/<int:pk>/<int:ppk>/', AddAssessment.as_view()),
17
18    path('to_do/', ToDo.as_view()),
19    path('to_do/<int:pk>/', ToDoDetail.as_view()),
20    path('to_do/<int:pk>/add_assignment/', AddAssignment.as_view()),
21    path('grades/', Statistics.as_view()),
22 ]
```

## views.py

```
class IndexView(TemplateView):
    template_name = "index.html"

class LoginFormView(FormView):
    form_class = AuthenticationForm
    template_name = "login.html"
    model = CustomUser
    success_url = "/profile"
    def form_valid(self, form):
        login(self.request, form.get_user())
        return super(LoginFormView, self).form_valid(form)

class TeacherRegisterFormView(FormView):
    form_class = TeacherCreationForm
    template_name = "register.html"
    model = CustomUser
    success_url = "/login/"
    def form_valid(self, form):
        form.save()
        return super(TeacherRegisterFormView, self).form_valid(form)

class StudentRegisterFormView(FormView):
    form_class = StudentCreationForm
    template_name = "register.html"
    model = CustomUser
    success_url = "/login/"
    def form_valid(self, form):
        form.save()
        return super(StudentRegisterFormView, self).form_valid(form)
```

## forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm, UserChangeForm
from .models import *

subjects = [
    ('Mathematics', 'Mathematics'),
    ('Physics', 'Physics'),
    ('Biology', 'Biology'),
    ('Informatics', 'Informatics')
]

group_numbers = [
    ('1', '1'),
    ('2', '2'),
    ('3', '3'),
    ('4', '4'),
]

roles = [
    ('teacher', 'учитель'),
    ('student', 'студент'),
]

class TeacherCreationForm(UserCreationForm):
    role = forms.Select(choices=roles)
    subject = forms.Select(choices=subjects)
    class Meta:
        model = CustomUser
        fields = [
            'username',
            'role',
            'subject',
            'name',
        ]

class StudentCreationForm(UserCreationForm):
    role = forms.Select(choices=roles)
    group_number = forms.Select(choices=group_numbers)
    class Meta:
        model = CustomUser
        fields = [
            'username',
            'role',
            'group_number',
            'name',
        ]

class AddHomework(forms.ModelForm):
    class Meta:
        model = Homework
        exclude = ['teacher']
        fields = ['group_number',
                  'subject',
                  'deadline',
                  'description']

class AddAssignment(forms.ModelForm):
    class Meta:
        model = Assignment
        exclude = ['student', 'homework']
        fields = ['solution']

class AddAssessment(forms.ModelForm):
    class Meta:
        model = Assessment
        exclude = ['student', 'homework']
        fields = ['grade']
```

<http://127.0.0.1:8000/>

[Войти](#)

[Зарегестировать учителя](#)

[Зарегестировать студента](#)

Войти в аккаунт

Username:

Password:

<http://127.0.0.1:8000/register/teacher>

Регистрация

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Role:  ▾

Subject:  ▾

Name:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

<http://127.0.0.1:8000/register/student>

## Регистрация

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/\_ only.

Role:  ▾

Group number:  ▾

Name:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

<http://127.0.0.1:8000/profile/>

[Выйти](#)

## Профиль учителя

Ivan Mathematics

[Добавить дз](#)

[Мое дз](#)

[Выйти](#)

## Профиль студента

Vasya 1

[Домашнее задание](#)

[Статистика](#)

Добавляем дз

[Выйти](#)

## Профиль учителя

teacher Mathematics

[Добавить дз](#)

[Мое дз](#)

Group number:  ▾

Subject:  ▾

Deadline:

дз по математике для 4 группы

Description:

[Назад](#)



[Выйти](#)

## Профиль учителя

teacher Mathematics

[Добавить дз](#)

[Мое дз](#)

---

[4 Mathematics](#) [Удалить](#)

---

Удаляем дз

[Выйти](#)

## Профиль учителя

teacher Mathematics

[Добавить дз](#)

[Мое дз](#)

Вы хотите удалить дз "4 Mathematics"?

Confirm

[Назад](#)



[Выйти](#)

## Профиль учителя

teacher Mathematics

[Добавить дз](#)

[Мое дз](#)

- 
- No tasks yet.
- 

Профиль студента

[Выйти](#)

## Профиль студента

student 4

[Домашнее задание](#)

[Статистика](#)

To do

[4 Mathematics](#)

---

Done

- Нет готовых дз

Можно нажать на задание и посмотреть его

---

## 4 Mathematics

- From: teacher

Deadline: Jan. 1, 2021

Description: дз по математике для 4 группы

[Добавить задание](#)

[Back](#)

Добавление ответа

### Профиль студента

student 4

[Домашнее задание](#)

[Статистика](#)

Solution:

сделано

Save

[Назад](#)

[Выйти](#)

## Профиль студента

student 4

[Домашнее задание](#)

[Статистика](#)

To do

- нет нового дз

Done

[4 Mathematics](#)

## Проверка учителем

[Выйти](#)

## Профиль учителя

teacher Mathematics

[Добавить дз](#)

[Мое дз](#)

4 Mathematics

- Deadline: Jan. 1, 2021

- student

Solution: сделано

[Check](#)

[Назад](#)

[Выйти](#)

## Профиль учителя

teacher Mathematics

[Добавить дз](#)

[Мое дз](#)

Grade:

Статистика ученика

[Выйти](#)

## **Профиль студента**

student 4

[Домашнее задание](#)  
[Статистика](#)

**Mathematics**

student 5

---

[Back](#)