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: <u>Реализация простого сайта средствами DJANGO</u>

Выполнил:

Студент: Барабанов Д.К., К33402

Проверил:

Преподаватель: Говоров А.И.

Санкт-Петербург

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

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

models.py

```
from django.db import models
from django.conf import settings
from django.contrib.auth.models import AbstractUser
    ('Mathematics', 'Mathematics'),
('Physics', 'Physics'),
('Biology', 'Biology'),
('Informatics', 'Informatics')
group_numbers = [
    ('1', '1'),
('2', '2'),
('3', '3'),
('4', '4'),
roles = [
('teacher', 'учитель'),
('student', 'студент'),
class CustomUser(AbstractUser):
  role = models.CharField(max_length=10, choices=roles, null=True)
   group_number = models.CharField(max_length=6, choices=group_numbers, null=True)
    subject = models.CharField(max_length=50, choices=subjects, null=True)
    name = models.CharField(max_length=100, null=True)
    def __str__(self):
    return '{}'.format(self.name)
 name = models.CharField(max_length=100, null=True)
subject = models.CharField(max_length=50, choices=subjects, null=True)
    def __str__(self):
    return '{}'.format(self.name)
 name = models.CharField(max_length=100, null=True)
group number = models.CharField(
    group_number = models.CharField(max_length=6, choices=group_numbers, null=True)
def __str__(self):
    return '{}'.format(self.name)
 group_number = models.CharField(max_length=6, choices=group_numbers, null=True)
subject = models.CharField(max_length=50, choices=subjects)
teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
    deadline = models.DateField()
    description = models.TextField(max_length=10000)
    def __str__(self):
    return '{} {}'.format(self.group_number, self.subject)
class Assignment(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE)
homework = models.ForeignKey(Homework, on_delete=models.CASCADE)
     solution = models.TextField(max_length=10000)
student = models.ForeignKey(Student, on_delete=models.CASCADE)
    homework = models.ForeignKey(Homework, on_delete=models.CASCADE)
    grade = models.CharField(max_length=2)
    def __str__(self):
           return '{} - {}'.format(self.student, self.homework.subject)
```

urls.py

```
from django.urls import path
    from .views import *
    urlpatterns = [
        path('', LoginFormView.as_view()),
        path('login/', LoginFormView.as_view()),
        path('register/teacher', TeacherRegisterFormView.as_view()),
        path('register/student', StudentRegisterFormView.as_view()),
        path('profile/', LoginPass),
        path('logout/', LogoutView.as_view()),
        path('homework/add/', AddHomework.as_view()),
        path('homework/<int:pk>/', HomeworkDetail.as_view(), name='teacher_detail'),
        path('homework/<int:pk>/delete/', DeleteHomework.as_view()),
        path('homeworks/', Homeworks.as_view()),
        path('homework/<int:pk>/<int:ppk>/', AddAssessment.as_view()),
        path('to_do/', ToDo.as_view()),
        path('to do/<int:pk>/', ToDoDetail.as view()),
        path('to_do/<int:pk>/add_assignment/',AddAssignment.as_view()),
        path('grades/', Statistics.as_view()),
    1
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 = [
    ('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)
         model = CustomUser
         fields = [
             'username',
class StudentCreationForm(UserCreationForm):
    role = forms.Select(choices=roles)
    group_number = forms.Select(choices=group_numbers)
         model = CustomUser
         fields = [
              'group_number',
class AddHomework(forms.ModelForm):
         model = Homework
         exclude = ['teacher']
         fields = ['group_number',
                     'subject',
                    'deadline',
                    'description']
class AddAssignment(forms.ModelForm):
    class Meta:
         model = Assignment
        exclude = ['student', 'homework']
fields = ['solution']
         model = Assessment
         exclude = ['student', 'homework']
         fields = ['grade']
```

http://127.0.0.1:8000/

<u>Войти</u>		
Зарегестировать учителя		
Зарегестировать студента		
Войти в аккаунт		
Username:		
Password:		
Войти		

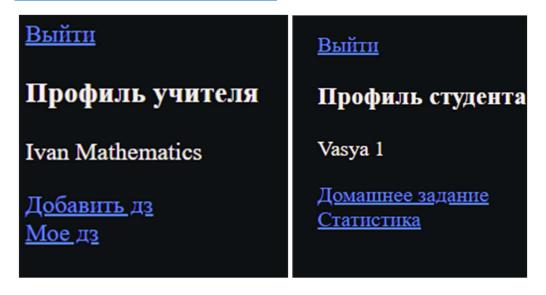
http://127.0.0.1:8000/register/teacher

Регистрация			
Username: Required. 1	50 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.		
Зарегистрироваться			

http://127.0.0.1:8000/register/student

Регистрация	
Username: Required. 15	60 characters or fewer. Letters, digits and @/./+/-/_ only.
Role:	
Group number:	
Name:	
Password:	
 Your password can't be too similar to your of Your password must contain at least 8 chara Your password can't be a commonly used password can't be entirely numeric. 	cters.
Password confirmation:	Enter the same password as before, for verification.
Зарегистрироваться	

http://127.0.0.1:8000/profile/



Добавляем дз

<u>Выйти</u>
Профиль учителя
teacher Mathematics
<u>Добавить дз</u> <u>Мое дз</u>
Group number: 4
Subject: Mathematics >
Deadline: 2021-01-01
дз по математике для 4 группы
Description:
Save
<u>Наза</u> д

Выйти Профиль учителя teacher Mathematics Добавить дз Мое дз 4 Mathematics Удалить

Удаляем дз

Выйти Профиль учителя teacher Mathematics Добавить дз Moe дз Вы хотите удалить дз "4 Mathematics"? Confirm Назад

<u>Выйти</u>

Профиль учителя

teacher Mathematics

<u>Добавить дз</u> <u>Мое дз</u>

No tasks yet.

Профиль студента

<u>Выйти</u>

Профиль студента

student 4

<u>Домашнее задание</u> <u>Статистика</u>

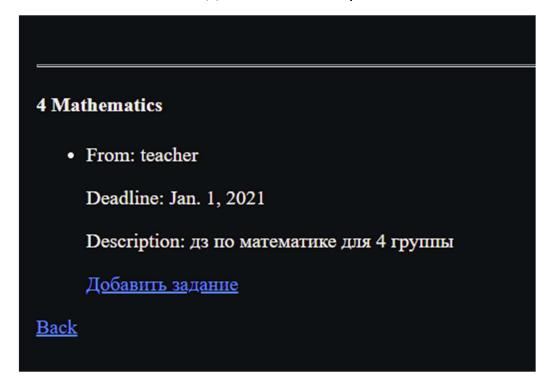
To do

4 Mathematics

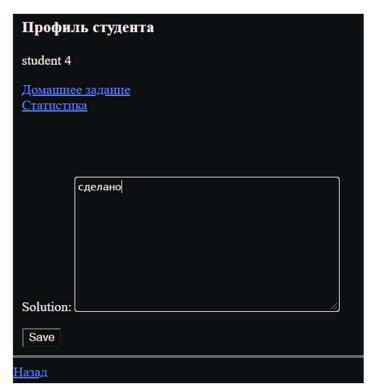
Done

• Нет готовых дз

Можно нажать на задание и посмотреть его

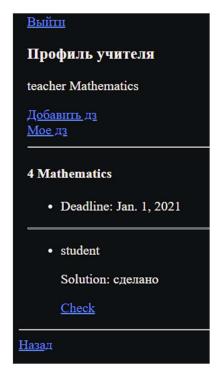


Добавление ответа





Проверка учителем



<u>Выйти</u>
Профиль учителя
teacher Mathematics
<u>Добавить дз</u> <u>Мое дз</u>
Grade: 5
Save

Статистика ученика

Выйти Профиль студента student 4 Домашнее задание Статистика Mathematics student 5 Back