

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

Факультет «Инфокоммуникационных технологий»

Направление подготовки «45.03.04 Интеллектуальные системы в гуманитарной сфере»

**О Т Ч Е Т**

Лабораторная работа №2

---

**Тема задания:** Реализация простого сайта средствами Django

---

**Выполнил:**

Студент Самощенко А.А. – К33421  
(Фамилия И.О.) номер группы

**Проверил:**

Преподаватель Говоров А.И.  
(Фамилия И.О.)

**Санкт-Петербург  
2020**

## Ход работы.

models.py:

```
from django.db import models
from django.contrib.auth.models import AbstractUser

class User(AbstractUser):
    homeworks = models.ManyToManyField('Homework', through='Answer')

class Homework(models.Model):
    course = models.CharField(max_length=100)
    teacher = models.CharField(max_length=80)
    date = models.DateField()
    deadline = models.DateField()
    description = models.TextField(max_length=10000)

    def __str__(self):
        return "{}: {}: {}".format(self.id, self.course, self.date)

class Answer(models.Model):
    student = models.ForeignKey(User, on_delete=models.CASCADE)
    homework = models.ForeignKey(Homework, on_delete=models.CASCADE)
    grade = models.CharField(max_length=3)
    answer = models.TextField(max_length=1000)
```

forms.py:

```
from django.contrib.auth.forms import UserCreationForm
from django import forms
from .models import User, Answer

class CreateUserForm(UserCreationForm):
    class Meta:
        model = User
        fields = [
            'username',
            'first_name',
            'last_name',
            'password1',
            'password2'
        ]

class AddAnswerForm(forms.ModelForm):
    class Meta:
        model = Answer
        fields = [
            'student',
            'homework',
            'answer'
        ]

    def __init__(self, *args, **kwargs):
        super(AddAnswerForm, self).__init__(*args, **kwargs)
        self.fields['student'].disabled = True
        self.fields['homework'].disabled = True
```

Регистрация в views.py:

```
def registerPage(request):
    if request.user.is_authenticated:
        return redirect('/homeworks/')
    else:
        form = CreateUserForm()
        if request.method == 'POST':
            form = CreateUserForm(request.POST)
            if form.is_valid():
                form.save()
                user = form.cleaned_data.get('username')
                messages.success(request, 'Account was created for ' + user)
                return redirect('login')
        context = {'form': form}
        return render(request, 'register.html', context)

def loginPage(request):
    if request.user.is_authenticated:
        return redirect('/homeworks/')
    else:
        if request.method == 'POST':
            username = request.POST.get('username')
            password = request.POST.get('password')
            user = authenticate(request, username=username, password=password)
            if user is not None:
                login(request, user)
                return redirect('/homeworks/')
            else:
                messages.info(request, 'Username OR password is incorrect')
        context = {}
        return render(request, 'login.html', context)

def logoutUser(request):
    logout(request)
    return redirect('login')
```

Homework в views.py:

```
class HomeworksList(ListView):
    model = Homework
    template_name = 'homeworks.html'
    queryset = model.objects.all()

    def get_queryset(self):
        task = self.request.GET.get('task')
        if task:
            try:
                task = int(task)
                queryset = self.queryset.filter(task=task)
            except ValueError:
                queryset = self.model.objects.none()
            return queryset
        return self.queryset

class HomeworkDetail(DetailView):
    model = Homework
    template_name = "homework_detail.html"
```

Answers в views.py:

```
class AnswersList(ListView):
    model = Answer
    template_name = 'answers.html'

    def get_queryset(self):
        return Answer.objects.all()

class AddAnswer(CreateView):
    form_class = AddAnswerForm
    model = Answer
    template_name = 'add_answer.html'

    def get_initial(self):
        initial = super(AddAnswer, self).get_initial()
        initial = initial.copy()
        initial['student'] = self.request.user.id
        initial['homework'] = get_object_or_404(Homework, pk=self.kwargs['pk'])
        return initial

class AnswerUpdate(UpdateView):
    model = Answer
    fields = ['answer']
    template_name = 'update_answer.html'

class AnswerDelete(DeleteView):
    model = Answer
    template_name = 'delete_answer.html'

class GradesTable(ListView):
    model = Answer
    template_name = 'grades_table.html'

    def get_queryset(self):
        return Answer.objects.all()
```

urls.py приложения:

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

urlpatterns = [
    path('register/', views.registerPage, name="register"),
    path('login/', views.loginPage, name='login'),
    path('logout/', views.logoutUser, name='logout'),
    path('homeworks/', views.Homeworkslist.as_view()),
    path('homeworks/<int:pk>', views HomeworkDetail.as_view()),
    path('answers/', views.Answerslist.as_view()),
    path('answers/<int:pk>/add/', views.AddAnswer.as_view(success_url="/answers/")),
    path('answers/<int:pk>/update/', views.AnswerUpdate.as_view(success_url="/answers/")),
    path('answers/<int:pk>/delete/', views.AnswerDelete.as_view(success_url="/answers/")),
    path('grades/', views.GradesTable.as_view()),
]
```

urls.py проекта:

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('hw_system.urls')),
]
```

admin.py:

```
from django.contrib import admin
from .models import *

admin.site.register(User)
admin.site.register(Homework)
admin.site.register(Answer)
```

Вход:

## LOG IN

No account? [Register](#)

Регистрация:

## Register Account

Have an account? [Login](#)

Список домашних заданий до их добавления:

# Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as Alex



[Quit](#)

---

## Homeworks:

- No homeworks for now.

Добавление домашнего задания через django-admin:

Course:	Mathematical Methods of ML
Teacher:	Ivanov Ivan Ivanovich
Date:	2020-10-10 Today 
Note: You are 3 hours ahead of server time.	
Deadline:	2020-10-24 Today 
Note: You are 3 hours ahead of server time.	
Description:	<div>Реализовать kNN.</div>

Список домашних заданий после добавления:

# Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as Alex

[Quit](#)

---

## Homeworks:

4:Mathematical Methods of ML:2020-10-10

**Course:** Mathematical Methods of ML

**Date:** Oct. 10, 2020

**Deadline:** Oct. 24, 2020

**Description:** Реализовать kNN.



Список решений до их добавления:

# Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as Alex

[Quit](#)

---

## Solutions:

- No solutions for now.

Добавление решения:

Signed in as Alex

[Quit](#)

---

## Add answer:

Student:

Homework:

Answer: 

Реализация kNN.

[Back to the answers](#)

Список решений после добавления решения:

# Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as Alex

[Quit](#)

---

## Solutions:

4:Mathematical Methods of ML:2020-10-10 [Change](#) [Delete](#)

**Solution:** Реализация kNN.

Попытка изменить:

Signed in as Alex

[Quit](#)

---

## Update answer:

Другая реализация kNN.

Answer:

[Back to the answers](#)

После изменения:

# Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as Alex

[Quit](#)

---

## Solutions:

4:Mathematical Methods of ML:2020-10-10 [Change](#) [Delete](#)

**Solution:** Другая реализация kNN.

Попытка удалить:

Signed in as Alex

[Quit](#)

---

## Delete answer:

Are you sure that you want to delete the solution to the "4:Mathematical Methods of ML:2020-10-10"?

Submit

[Back to the answers](#)

Таблица успеваемости до выставления оценки преподавателем-администратором:

## Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as Alex

[Quit](#)

Student	Course	Homework	Grade
Aleksei Alekseev	Mathematical Methods of ML	<a href="#">4:Mathematical Methods of ML:2020-10-10</a>	

Добавление оценки:

Change answer

HISTORY

Student:

Alex



Homework:

4:Mathematical Methods of ML:2020-10-10



Grade:

5

Answer:

Другая реализация kNN.

Таблица с оценкой:

## Homework Manager

- [Homeworks](#)
- [Answers](#)
- [Grades](#)

Signed in as admin

[Quit](#)

Student	Course	Homework	Grade
Aleksei Alekseev	Mathematical Methods of ML	<a href="#">4:Mathematical Methods of ML:2020-10-10</a>	5