

Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»



Лабораторные работы по курсу:
«Разработка Интернет Приложений»

ЛР7. Авторизация, работа с формами и Django Admin

Исполнитель:

Студент группы РТ5-51

Умряев Д.Т.

Преподаватель:

Гапанюк Ю. Е.

«__» _____



Москва 2017 г.

Задание и порядок выполнения

Основная цель данной лабораторной работы – научиться обрабатывать веб-формы на стороне приложения, освоить инструменты, которые предоставляет Django, по работе с формами. Также в этой лабораторной работе необходимо освоить инструменты Django по работе с авторизацией и реализовать простейшую авторизацию. Напоследок, необходимо познакомиться с инструментом администрирования Django – как в несколько строчек кода сделать панель администратора сайта.

1. Создайте view, которая возвращает форму для регистрации.

Поля формы:

- Логин
- Пароль
- Повторный ввод пароля
- Email
- Фамилия
- Имя

2. Создайте view, которая возвращает форму для авторизации.

Поля формы:

- Логин
- Пароль

3. При отправке формы регистрации во view проверять каждый параметр по правилам валидации, если валидация всех полей пройдена, то создавать пользователя и делать перенаправление на страницу логина, а ошибки, если они есть, выводить над формой.

Правила валидации:

- Логин не меньше 5 символов
- Пароль не меньше 8 символов
- Пароли должны совпадать
- Все поля должны быть заполнены
- Логин – уникален для каждого пользователя

4. При возникновении ошибок в момент отправки формы, введенные значения в полях ввода, кроме пароля, не должны исчезать.

5. Переписать view регистрации с использованием Django Form, правила валидации удалить из view, использовать встроенный механизм валидации полей.

6. Во view авторизации реализовать логин при POST запросе. При успешной авторизации должен происходить переход на страницу успешной авторизации.

7. Страница успешной авторизации должна проверять, что пользователь авторизован. Иначе делать перенаправление на страницу авторизации.

8. Реализовать view для выхода из аккаунта.

9. Заменить проверку на авторизацию на декоратор login_required

10. Добавить superuser'a через команду manage.py

11. Подключить django.contrib.admin и войти в панель администрирования.

12. Зарегистрировать все свои модели в django.contrib.admin

13. Для выбранной модели настроить страницу администрирования:

- Настроить вывод необходимых полей в списке
- Добавить фильтры
- Добавить поиск
- Добавить дополнительное поле в список

Исходный код:

settings.py

```
"""
```

Django settings for lab7 project.

Generated by 'django-admin startproject' using Django 1.11.6.

For more information on this file, see

<https://docs.djangoproject.com/en/1.11/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/1.11/ref/settings/>

```
"""
```

```
import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
```

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY = 'nedg9xz7)a@#^n@-wt6v7%h5r3bzu9=33kf-a5q($##sfoq^!('
```

```
# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [
```

```
    'django.contrib.admin',
```

```
    'django.contrib.auth',
```

```
    'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
```

```
    'django.contrib.messages',
```

```
    'django.contrib.staticfiles',
```

```
    # 'tutoring.apps.MyAppConfig',
```

```
    'tutoring',
```

```
]
```

```
MIDDLEWARE = [
```

```
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'lab7.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')]
    },
    'APP_DIRS': True,
    'OPTIONS': {
        'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
            'django.contrib.messages.context_processors.messages',
        ],
    },
],
]
```

```
WSGI_APPLICATION = 'lab7.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases
```

```
# DATABASES = {
#     'default': {
#         'ENGINE': 'django.db.backends.sqlite3',
#         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
#     }
# }
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'tutoring_django',
        'USER': 'dbuser',
        'PASSWORD': '123',
        'HOST': 'localhost',
```

```
'PORT': 3306, # Стандартный порт MySQL
'OPTIONS': {'charset': 'utf8'},
'TEST_CHARSET': 'utf8',
}
}
```

Password validation

<https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators>

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

Internationalization

<https://docs.djangoproject.com/en/1.11/topics/i18n/>

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

Static files (CSS, JavaScript, Images)

<https://docs.djangoproject.com/en/1.11/howto/static-files/>

```
STATIC_URL = '/static/'
```

```
LOGIN_URL = '/login/'
```

lab7/urls.py

```
"""lab7 URL Configuration
```

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/1.11/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `url(r'^$', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `url(r'^$', Home.as_view(), name='home')`

Including another `URLconf`

1. Import the `include()` function: `from django.conf.urls import url, include`
2. Add a URL to `urlpatterns`: `url(r'^blog/', include('blog.urls'))`

```
"""
```

```
from django.conf.urls import url, include
from django.contrib import admin
from tutoring.views import *
from django.conf.urls.static import static
from django.conf import settings
```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^db/universities', UniversitiesList.as_view(), name='universities_url'),
    url(r'^db/regions', RegionsList.as_view(), name='regions_url'),
    url(r'^db/subjects', SubjectsList.as_view(), name='subjects_url'),
    url(r'^db/tutors', TutorsList.as_view(), name='tutors_url'),
    url(r'^db/', db, name='db_url'),
    url(r'^orders/', include('tutoring.urls')),
    url(r'^orders/', OrdersView.as_view(), name='orders_url'),
    url(r'^main/', main, name='main_url'),
    url(r'^registration/', registration, name='registration_url'),
    url(r'^registration2/', registration2, name='registration2_url'),
    url(r'^login/', login, name='login_url'),
    url(r'^success/', success, name='success_url'),
    url(r'^logout/', logout, name='logout_url'),
    url(r'^(?P<prog_lang>\w+)/', prog_lang, name='prog_lang_url'),
    url(r'^.*$', RedirectView.as_view(url='/main/', permanent=False), name='index'),
```

```
]
```

```
if settings.DEBUG:
```

```
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_URL)
```

views.py

```
from django.shortcuts import render, HttpResponseRedirect
```

```
from django.views import View
from django.views.generic import ListView
from datetime import datetime
from tutoring.models import Education, Subjects, Regions, Tutors
from tutoring.registration import *
from django.contrib.auth.models import User
from django.contrib import auth
from django.contrib.auth import authenticate
from django.contrib.auth.decorators import login_required
# Create your views here.
```

```
class OrdersView(View):
    def get(self, request):
        variable = 'Django'
        today_date = datetime.now()
        data = {
            'orders': [
                {'title': 'Первый заказ', 'id': 1},
                {'title': 'Второй заказ', 'id': 2},
                {'title': 'Третий заказ', 'id': 3}
            ]
        }
        return render(request, 'orders.html', locals())
```

```
class OrderView(View):
    def get(self, request, id):
        variable = 'Django'
        today_date = datetime.now()
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', locals())
```

```
def main(request):
    return render(request, 'main.html', locals())
```

```
def prog_lang(request, prog_lang):
    name = ['C++', 'Python', 'Java']
    cpp_info = 'C++ — компилируемый, статически типизированный язык программирования
общего назначения. Синтаксис C++ унаследован от языка C. Одним из принципов разработки
было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле
надмножеством C; множество программ, которые могут одинаково успешно транслироваться как
```

компиляторами С, так и компиляторами С++, довольно велико, но не включает все возможные программы на С.'

java_info = 'Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.'

python_info = 'Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.'

```
info = [cpp_info, java_info, python_info]
data1 = {'lang': {'id': prog_lang}}
data2 = {'langs': [{'id': 'cpp', 'lang_name': 'C++', 'info': cpp_info},
                  {'id': 'java', 'lang_name': 'Java', 'info': java_info},
                  {'id': 'python', 'lang_name': 'Python', 'info': python_info}]}
return render(request, 'prog_lang.html', locals())
```

```
def db(request):
    return render(request, 'db.html', locals())
```

```
class UniversitiesList(ListView):
    model = Education
    template_name = "universities.html"
```

```
class SubjectsList(ListView):
    model = Subjects
    template_name = "subjects.html"
```

```
class RegionsList(ListView):
    model = Regions
    template_name = "regions.html"
```

```
class TutorsList(ListView):
    model = Tutors
    template_name = "tutors.html"
```

```
def registration(request):
    errors = {'username': "", 'password': "", 'password2': "", 'email': "", 'firstname': "", 'surname': ""}
    error_flag = False
    if request.method == 'POST':
        username = request.POST.get('username')
```



```

if not username:
    errors['username'] = 'Введите логин'
    error_flag = True
elif len(username) < 5:
    errors['username'] = 'Логин должен превышать 5 символов'
    error_flag = True
elif User.objects.filter(username=username).exists():
    errors['username'] = 'Такой логин уже существует'
    error_flag = True
password = request.POST.get('password')
if not password:
    errors['password'] = 'Введите пароль'
    error_flag = True
elif len(password) < 8:
    errors['password'] = 'Длина пароля должна превышать 8 символов'
password_repeat = request.POST.get('password2')
if password != password_repeat:
    errors['password2'] = 'Пароли должны совпадать'
    error_flag = True
email = request.POST.get('email')
if not email:
    errors['email'] = 'Введите e-mail'
firstname = request.POST.get('firstname')
if not firstname:
    errors['firstname'] = 'Введите имя'
surname = request.POST.get('surname')
if not surname:
    errors['surname'] = 'Введите фамилию'
if not error_flag:
    # ...
    user = User.objects.create_user(username=username, password=password, email=email,
first_name=firstname, last_name=surname)
    return HttpResponseRedirect('/login/')
return render(request, 'registration.html', locals())

```

```

def login(request):
    error = ""
    username = None
    password = None
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(username=username, password=password)
        if user:
            auth.login(request, user)
            return HttpResponseRedirect('/success/')
        else:

```

```

        error = "Пользователь не найден"
    return render(request, 'login.html', locals())

@login_required()
def success(request):
    # if not request.user.is_authenticated:
    #     return HttpResponseRedirect('/login/')
    return render(request, 'success.html', locals())

def logout(request):
    auth.logout(request)
    return HttpResponseRedirect('/main/')

def registration2(request):
    form = RegistrationForm(request.POST or None)
    if request.method == 'POST':
        if form.is_valid():
            user = User.objects.create_user(username=request.POST.get('username'),
                                             email=request.POST.get('email'),
                                             password=request.POST.get('password'),
                                             first_name=request.POST.get('firstname'),
                                             last_name=request.POST.get('surname'))

            # ...
            return HttpResponseRedirect('/login/')
        else:
            form = RegistrationForm()
    return render(request, 'registration2.html', {'form': form})

```

models.py

```

from django.db import models

# Create your models here.

class Education(models.Model):
    id_university = models.AutoField(primary_key=True)
    name_university = models.CharField(max_length=100, verbose_name='Университет')

    def __str__(self):
        return self.name_university

class Meta:
    verbose_name_plural = "Университеты"
    verbose_name = "Университет"

```

```

class Subjects(models.Model):
    id_subject = models.AutoField(primary_key=True)
    name_subject = models.CharField(max_length=100, verbose_name='Предмет')

    def __str__(self):
        return self.name_subject

    class Meta:
        verbose_name_plural = "Предметы"
        verbose_name = "Предмет"

class Regions(models.Model):
    id_region = models.AutoField(primary_key=True)
    name_region = models.CharField(max_length=100, verbose_name='Регион')

    def __str__(self):
        return self.name_region

    class Meta:
        verbose_name_plural = "Регионы"
        verbose_name = "Регион"

class Tutors(models.Model):
    id_tutor = models.AutoField(primary_key=True)
    name = models.CharField(max_length=45, verbose_name='Имя')
    surname = models.CharField(max_length=45, verbose_name='Фамилия')
    patronymic = models.CharField(max_length=45, verbose_name='Отчество')
    email = models.EmailField(max_length=254, verbose_name='E-mail')
    tel = models.CharField(max_length=20, verbose_name='Номер телефона')
    birth_date = models.DateField(verbose_name='Дата рождения')
    date_tutoring_begin = models.DateField(verbose_name='Дата начала преподавания')
    address = models.CharField(max_length=100, verbose_name='Адрес')
    region = models.ForeignKey(Regions, on_delete=models.CASCADE, verbose_name='Регион')
    edu = models.ManyToManyField(Education, verbose_name='Образование')
    subjects = models.ManyToManyField(Subjects, verbose_name='Преподаваемые предметы')

    def __str__(self):
        return '%s %s %s' % (self.name, self.surname, self.patronymic)

    def get_edu(self):
        return "\n".join([e.name_university for e in self.edu.all()])
    get_edu.short_description = 'Образование'

    def get_subjects(self):

```

```
return ", ".join([s.name_subject for s in self.subjects.all()])
get_subjects.short_description = 'Преподаваемые предметы'
```

```
class Meta:
    verbose_name_plural = "Репетиторы"
    verbose_name = "Репетитор"
```

registration.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
```

```
class RegistrationForm(forms.Form):
    username = forms.CharField(min_length=5, label='Логин')
    password = forms.CharField(min_length=6, widget=forms.PasswordInput, label='Пароль')
    password2 = forms.CharField(min_length=6, widget=forms.PasswordInput, label='Повторите
пароль')
    email = forms.EmailField(widget=forms.EmailInput, label='E-mail')
    firstname = forms.CharField(label='Имя')
    surname = forms.CharField(label='Фамилия')
```

apps.py

```
from django.apps import AppConfig
```

```
class MyAppConfig(AppConfig):
    name = 'tutoring'
    verbose_name = 'Репетиторство'
```

admin.py

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

```
# Register your models here.
```

```
class TutorsAdmin(admin.ModelAdmin):
    fields = (('name', 'surname', 'patronymic'), 'email', 'tel', 'birth_date', 'date_tutoring_begin', 'address',
            'region', 'edu', 'subjects')
    list_filter = ('region', ('edu', admin.RelatedOnlyFieldListFilter), ('subjects',
admin.RelatedOnlyFieldListFilter),)
    list_display = ('surname', 'name', 'patronymic', 'email', 'tel', 'birth_date', 'date_tutoring_begin', 'address',
            'region', 'get_edu', 'get_subjects')
    search_fields = ('name', 'surname', 'patronymic')
    list_per_page = 10
```

```

class SubjectsAdmin(admin.ModelAdmin):
    list_per_page = 10

admin.site.register(Education)
admin.site.register(Subjects, SubjectsAdmin)
admin.site.register(Regions)
admin.site.register(Tutors, TutorsAdmin)

admin.site.site_url = '/main/'
admin.site.site_header = 'Django Администрирование'
admin.site.index_title = 'Администрирование'

```

__init.py__

```
default_app_config = 'tutoring.apps.MyAppConfig'
```

base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Главная</title>
    {% load static %}
    <link rel="shortcut icon" href="{% static 'images/favicon.ico' %}" type="image/x-icon">
    <link rel="stylesheet" href="{% static 'Bootstrap/css/bootstrap.css' %}">
    <link rel="stylesheet" href="{% static 'Bootstrap/css/font-awesome.min.css' %}">
    <link type="text/css" rel="stylesheet" href="{% static 'Bootstrap/css/my.css' %}">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script type="text/javascript" src="{% static 'Bootstrap/js/bootstrap.min.js' %}"></script>
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">MYA<i class="fa fa-rub" aria-hidden="true"></i><i
class="fa fa-rub" aria-hidden="true"></i></a>
            </div>

```

```

<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav navbar-right">
    <li><a href="{% url 'main_url' %}">Домой</a></li>
    <li><a href="{% url 'orders_url' %}">Задание</a></li>
    <li><a href="{% url 'db_url' %}">База Данных</a></li>
    <li>
      <div class="dropdown">
        <button class="dropdown-toggle" data-toggle="dropdown">
          
          <span class="caret"></span>#}
        </button>
        <div class="dropdown-content">
          {% if request.user.is_authenticated % }
            <span>{{ request.user.username }}</span>
            <a href="{% url 'logout_url' %}">Выйти</a>
          {% else % }
            <a href="{% url 'login_url' %}">Вход</a>
            <a href="{% url 'registration_url' %}">Регистрация</a>
            <a href="{% url 'registration2_url' %}">Регистрация2</a>#}
          {% endif % }
        </div>
      </div>
    </li>
  </ul>
</div>
</div>
{% block body1 %}{% endblock %}
</body>
</html>

```

main.html

```

{% extends 'base.html' %}

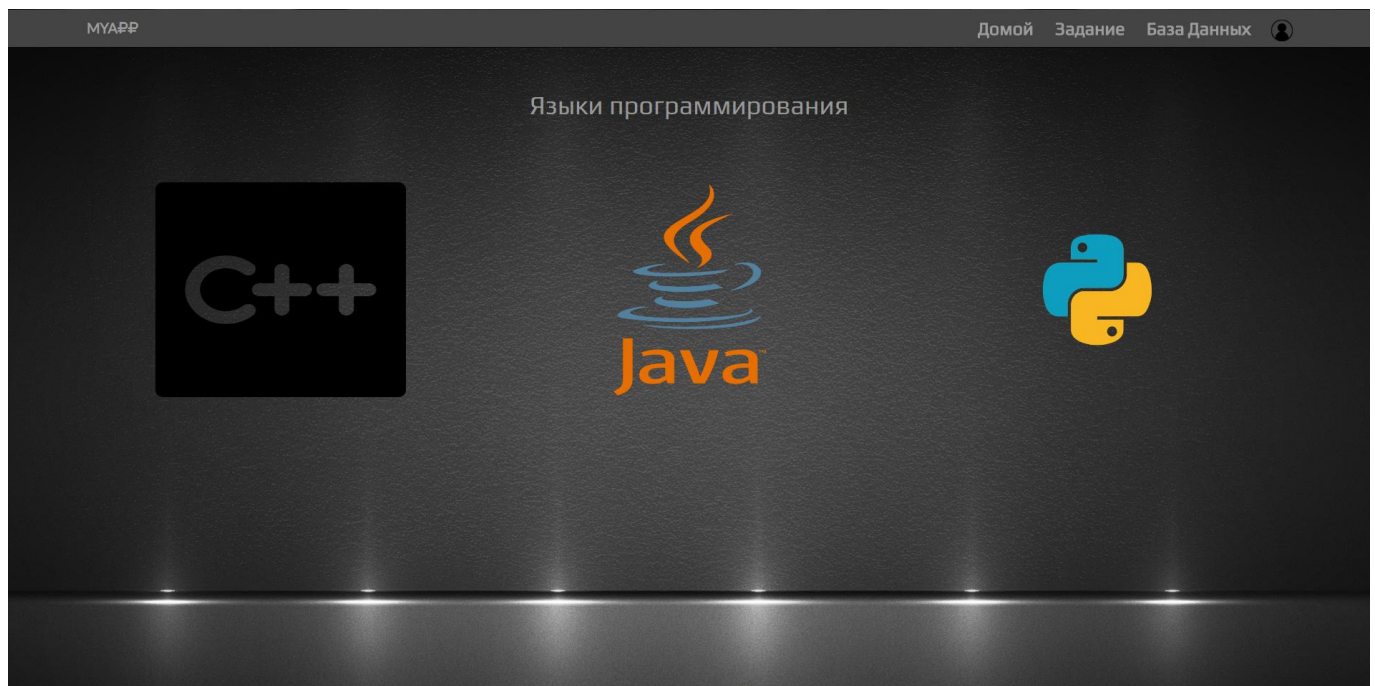
{% block body1 %}
  <div id="header1">
    <div class="container">
      <div class="row centered">
        <div class="col-lg-8 col-lg-offset-2">
          <h1>Programming Languages</h1>
        </div>
      </div>
    </div>
    <br><br>
    <div class="container">
      <div class="row centered">
        {% load static %}
        <div class="col-lg-4">

```

```

        <a href="{% url 'prog_lang_url' 1 %}"></a>
    </div>
    <div class="col-lg-4">
        <a href="{% url 'prog_lang_url' 2 %}"></a>
    </div>
    <div class="col-lg-4">
        <a href="{% url 'prog_lang_url' 3 %}"></a>
    </div>
</div>
</div>
</div>
{% endblock %}

```



db.html

```

{% extends 'base.html' %}

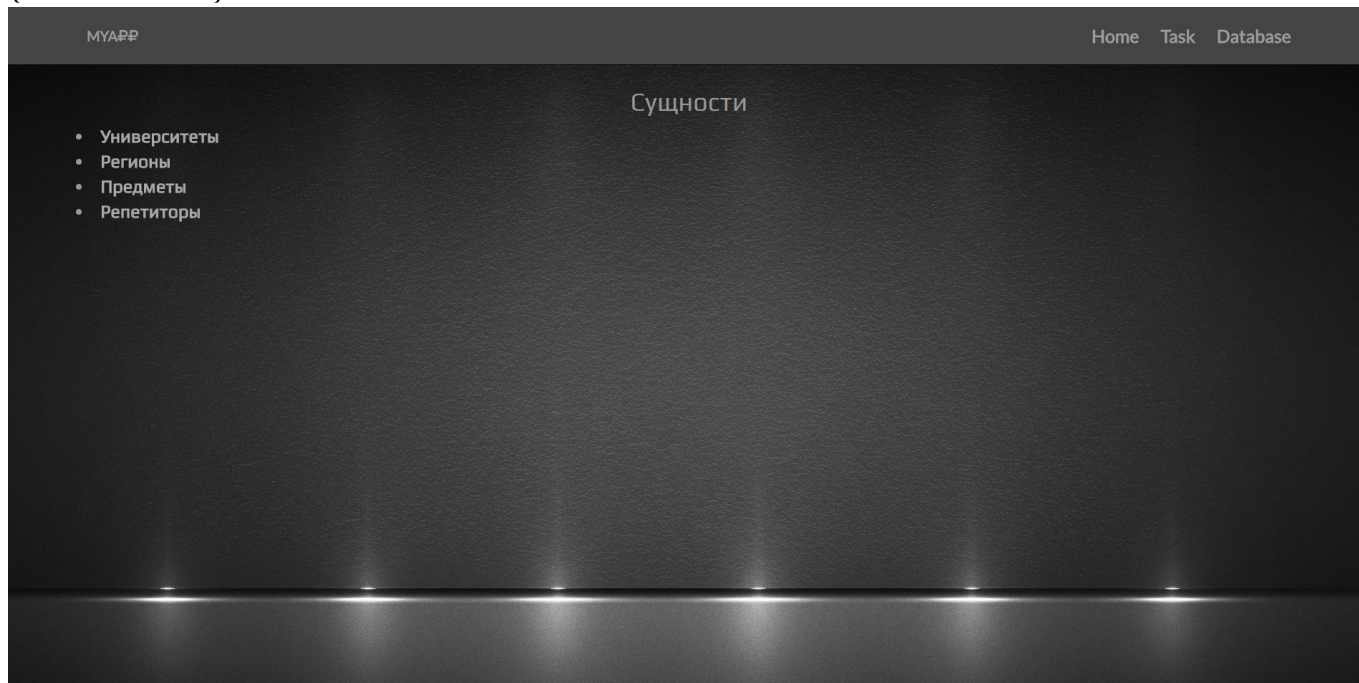
{% block body1 %}
    <div id="header1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h1><br><br>Сущности</h1>
                </div>
            </div>
        </div>
    </div>
    <div class="container">

```

```

<div class="row centered">
  <li><a href="{% url 'universities_url' %}">Университеты</a></li>
  <li><a href="{% url 'regions_url' %}">Регионы</a></li>
  <li><a href="{% url 'subjects_url' %}">Предметы</a></li>
  <li><a href="{% url 'tutors_url' %}">Репетиторы</a></li>
</div>
</div>
</div>
{% endblock %}

```



universitites.html

```

{% extends 'base.html' %}

{% block body1 %}
<div id="header1">
  <div class="container">
    <div class="row centered">
      <div class="col-lg-8 col-lg-offset-2">
        <h1></h1>
      </div>
    </div>
  </div>
<br><br>
  <div class="container">
    <div class="row centered">
      <table>
        <tr>
          <th>ID</th>
          <th>Университет</th>
        </tr>

```



```

        {% for unvier in object_list %}
        <tr>
            <td>{{ unvier.id_university }}</td>
            <td>{{ unvier.name_university }}</td>
        </tr>
        {% endfor %}
    </table>
</div>
</div>
</div>
{% endblock %}

```

MYAPP		Home Task Database	
ID	Университет		
1	МГТУ им. Н.Э. Баумана		
2	МГУ им. Ломоносова		

regions.html

```

{% extends 'base.html' %}

{% block body1 %}
    <div id="header1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h1></h1>
                </div>
            </div>
        </div>
        <br><br>
        <div class="container">
            <div class="row centered">
                <table>
                    <tr>

```

```

        <th>ID</th>
        <th>Регион</th>
    </tr>
    {% for region in object_list %}
        <tr>
            <td>{{ region.id_region }}</td>
            <td>{{ region.name_region }}</td>
        </tr>
    {% endfor %}
</table>
</div>
</div>
{% endblock %}

```

MYAPP

[Home](#)
[Task](#)
[Database](#)

ID	Регион
1	Москва
2	Пермь
3	Владивосток
4	Волгоград
5	Санкт-Петербург
6	Уфа
7	Краснодар

subjects.html

```

{% extends 'base.html' %}

{% block body1 %}
    <div id="header1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h1></h1>
                </div>
            </div>
        </div>
    </div>
    <br><br>
    <div class="container">

```

```

<div class="row centered">
<table>
  <tr>
    <th>ID</th>
    <th>Предмет</th>
  </tr>
  {% for subject in object_list %}
    <tr>
      <td>{{ subject.id_subject }}</td>
      <td>{{ subject.name_subject }}</td>
    </tr>
  {% endfor %}
</table>
</div>
</div>
{% endblock %}

```

MYAPP		Home Task Database	
ID	Предмет		
1	Математика		
2	Физика		
3	Английский язык		
4	Химия		
5	Русский язык		
6	Обществознание		
7	Биология		
8	История		
9	Информатика		

tutors.html

```

{% extends 'base.html' %}

{% block body1 %}
  <div id="header1">
    <div class="container">
      <div class="row centered">
        <div class="col-lg-8 col-lg-offset-2">
          <h1></h1>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
<br><br>
<div class="container">
  <div class="row centered">
    <table>
      <tr>
        <th>ID</th>
        <th>Имя</th>
        <th>Фамилия</th>
        <th>Отчество</th>
        <th>Почта</th>
        <th>Телефон</th>
        <th>Дата рождения</th>
        <th>Дата начала преподавания</th>
        <th>Адрес</th>
        <th>Регион</th>
      </tr>
      {% for tutor in object_list %}
        <tr>
          <td>{{ tutor.id_tutor }}</td>
          <td>{{ tutor.name }}</td>
          <td>{{ tutor.surname }}</td>
          <td>{{ tutor.patronymic }}</td>
          <td>{{ tutor.email }}</td>
          <td>{{ tutor.tel }}</td>
          <td>{{ tutor.birth_date }}</td>
          <td>{{ tutor.date_tutoring_begin }}</td>
          <td>{{ tutor.address }}</td>
          <td>{{ tutor.region.name_region }}</td>
        </tr>
      {% endfor %}
    </table>
  </div>
</div>
{% endblock %}

```

ID	Имя	Фамилия	Отчество	Почта	Телефон	Дата рождения	Дата начала преподавания	Адрес	Регион
1	Иван	Иванов	Иванович	ivanov@mail.ru	89299993334	July 8, 1966	June 6, 1990	Пушкинская, 24	Москва
2	Петр	Петров	Петрович	retrov@mail.ru	89299993335	May 8, 1968	Feb. 6, 1989	Пушкинская, 64	Москва

login.html

```
{% extends 'base.html' %}

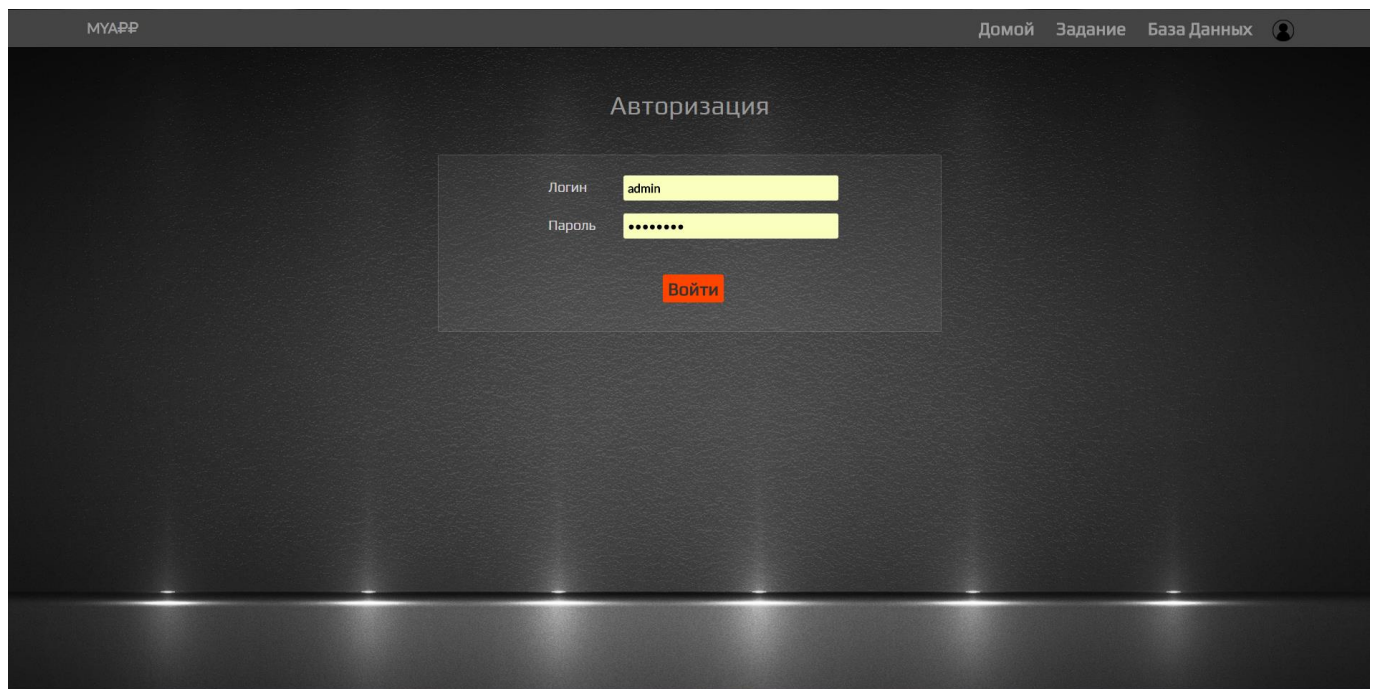
{% block body1 %}
<div id="body1">
  <div class="container">
    <div class="row centered">
      <div class="col-lg-8 col-lg-offset-2">
        <h1><br>Авторизация<br><br></h1>
      </div>
    </div>
  </div>
  <form method="POST">
    <div class="auth_block">
      {% csrf_token %}
      <div class="auth_block">
        <label class="login">
          Логин
        </label>
        <div class="form-block">
          <input type="text" name="username">
        </div>
      </div>
      <div class="auth_block">
        <label class="login">
          Пароль
        </label>
        <div class="form-block">
          <input type="password" name="password">
        </div>
      </div>
    </div>
  </form>
</div>
{% endblock %}
```



```

        </div>
    </div>
    {% if error != "" %}
        <p>{{ error }}</p>
    {% endif %}
    <br>
    <button type="submit">Войти</button>
</div>
</form>
</div>
{% endblock %}

```



registration.html

```

{% extends 'base.html' %}

{% block body1 %}
    <div id="body1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h1><br>Регистрация<br><br></h1>
                </div>
            </div>
        </div>
        <form method="POST">
            <div class="auth_block">
                {% csrf_token %}
                <div class="auth_blank">
                    <label>

```

```

    Логин
</label>
<div class="form-block">
    <input type="text" name="username" value="{{ username }}">
    {% if errors.username != " " %}
        <div class="form-error">{{ errors.username }}</div>
    {% endif %}
</div>
</div>
<div class="auth_blank">
    <label>
        Пароль
    </label>
    <div class="form-block">
        <input type="password" name="password">
        {% if errors.password != " " %}
            <div class="form-error">{{ errors.password }}</div>
        {% endif %}
    </div>
</div>
<div class="auth_blank">
    <label>
        Повторите пароль
    </label>
    <div class="form-block">
        <input type="password" name="password2">
        {% if errors.password2 != " " %}
            <div class="form-error">{{ errors.password2 }}</div>
        {% endif %}
    </div>
</div>
<div class="auth_blank">
    <label>
        E-mail
    </label>
    <div class="form-block">
        <input type="email" name="email" value="{{ email }}">
        {% if errors.email != " " %}
            <div class="form-error">{{ errors.email }}</div>
        {% endif %}
    </div>
</div>
<div class="auth_blank">
    <label>
        Имя
    </label>
    <div class="form-block">
        <input type="text" name="firstname" value="{{ firstname }}">

```

```

        {% if errors.firstname != " " %}
            <div class="form-error">{{ errors.firstname }}</div>
        {% endif %}
    </div>
</div>
<div class="auth_blank">
    <label>
        Фамилия
    </label>
    <div class="form-block">
        <input type="text" name="surname" value="{{ surname }}">
        {% if errors.surname != " " %}
            <div class="form-error">{{ errors.surname }}</div>
        {% endif %}
    </div>
</div>
<br>
<button type="submit">Зарегистрироваться</button>
</div>
</form>
<a style="font-size: 20px; font-weight: 100; text-decoration: underline;" href="{% url
'registration2_url' %}">Form version</a>
</div>
{% endblock %}

```

МУАРР Домой Задание База Данных

Регистрация

Логин	<input type="text"/>
Пароль	<input type="password"/>
Повторите пароль	<input type="password"/>
E-mail	<input type="text"/>
Имя	<input type="text"/>
Фамилия	<input type="text"/>

Зарегистрироваться

[Form version](#)

registration2.html

```

{% extends 'base.html' %}

{% block body1 %}

```



```

<div id="body1">
  <div class="container">
    <div class="row centered">
      <div class="col-lg-8 col-lg-offset-2">
        <h1><br>Регистрация 2<br><br></h1>
      </div>
    </div>
  </div>
  <form method= "POST">
    <div class="auth_block">
      { % csrf_token % }
      { { form.as_p } }
      <br>
      <button type= "submit"> Зарегистрироваться </button>
    </div>
  </form>
</div>
{ % endblock % }

```

Скриншот веб-интерфейса с темной темой. Вверху панель навигации с ссылками: «Домой», «Задание», «База Данных» и иконка пользователя. В центре экрана заголовок «Регистрация 2» над формой. Форма имеет следующие поля: «Логин:», «Пароль:», «Повторите пароль:», «E-mail:», «Имя:» и «Фамилия:», каждое с соответствующим текстовым полем ввода. Внизу формы находится красная кнопка «Зарегистрироваться».

success.html

```

{ % extends 'base.html' % }

{ % block body1 % }
  <div id="body1">
    <div class="container">
      <div class="row centered">
        <div class="col-lg-8 col-lg-offset-2">
          <h1><br>Вы успешно авторизованы!<br></h1>
          <h2>Через 5 секунд вы будете перенаправлены на главную страницу<br></h2>

```

```
<meta http-equiv="refresh" content="5; url=/main/">
</div>
</div>
</div>
{ % endblock % }
```

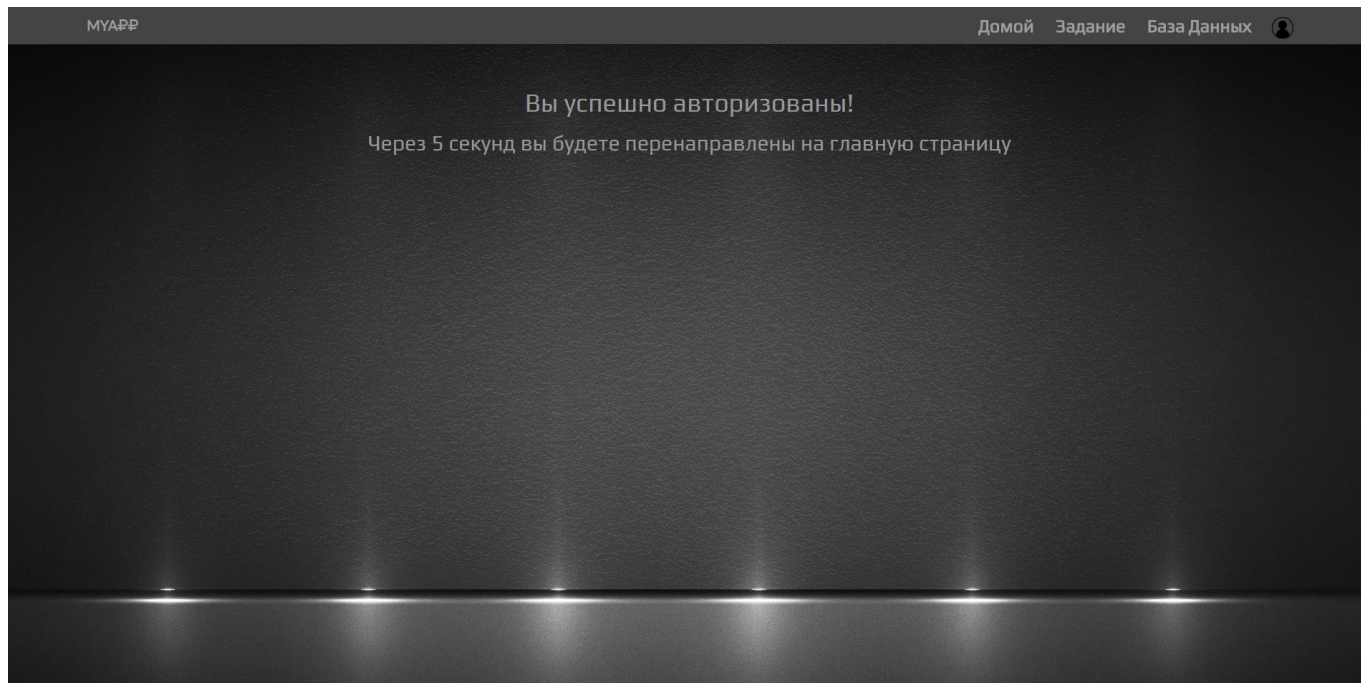
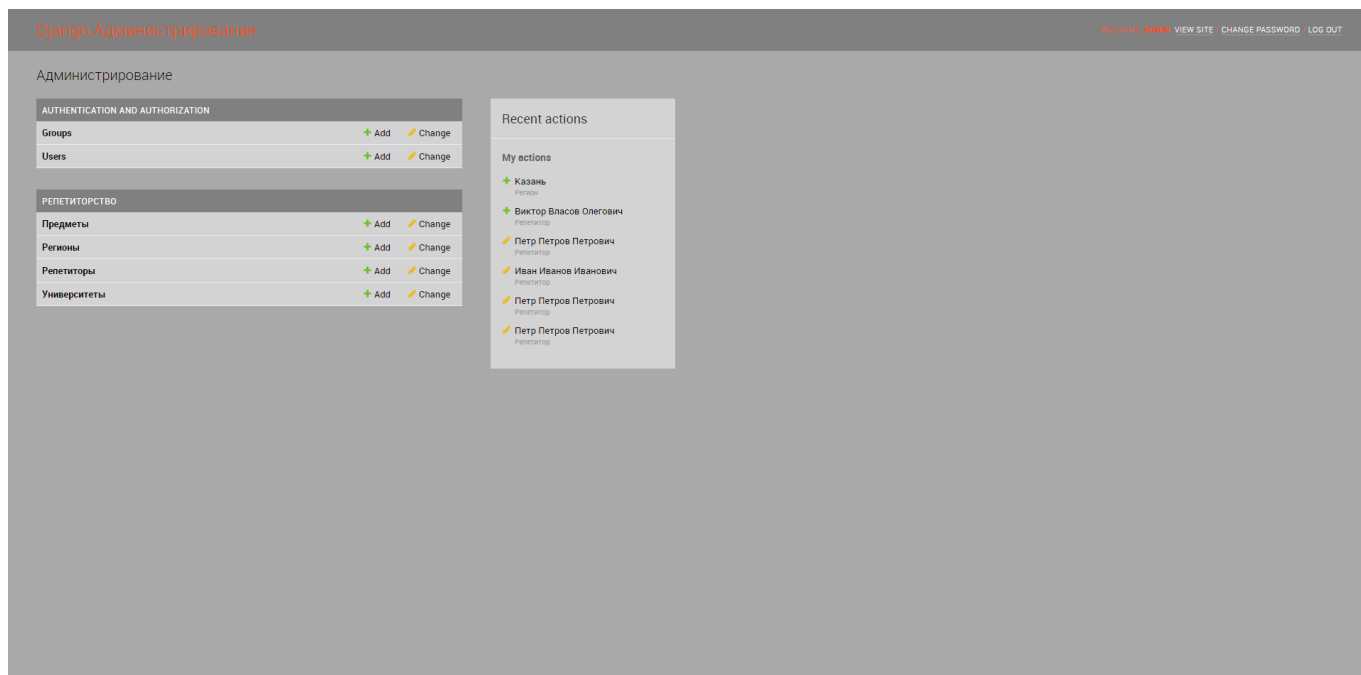


Фото из админки:



Q Search

Action: ----- Go 0 of 3 selected

	ФАМИЛИЯ	ИМЯ	ОТЕЧЕСТВО	E-MAIL	НОМЕР ТЕЛЕФОНА	ДАТА РОЖДЕНИЯ	ДАТА НАЧАЛА ПРЕПОДАВАНИЯ	АДРЕС	РЕГИОН	ОБРАЗОВАНИЕ	ПРЕПОДАВАЕМЫЕ ПРЕДМЕТЫ
<input type="checkbox"/>	Власов	Виктор	Олегович	vlasov@yandex.ru	89324214433	Dec. 22, 1970	Feb. 13, 1999	Московская, 54	Москва	МГУ им. Ломоносова	Математика, Физика, Информатика
<input type="checkbox"/>	Петров	Петр	Петрович	retrov@mail.ru	89299993335	May 8, 1968	Feb. 6, 1989	Пушкинская, 64	Москва	МГТУ им. Н.Э. Баумана	Математика, Физика, Английский язык, Химия, Русский язык
<input type="checkbox"/>	Иванов	Иван	Иванович	ivanov@mail.ru	89299993334	July 8, 1966	June 6, 1990	Пушкинская, 24	Москва	МГТУ им. Н.Э. Баумана	Русский язык

3 Репетиторы

FILTER

By Region

All
Москва
Пермь
Владивосток
Волгоград
Санкт-Петербург
Уфа
Краснодар
Казань

Ву Образование

All
МГТУ им. Н.Э. Баумана
МГУ им. Ломоносова
-

Ву Преподаваемые предметы

All
Математика
Физика
Английский язык
Химия
Русский язык
Информатика