

**Московский государственный технический  
университет им. Н.Э. Баумана.**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5. Курс «Разработка интернет приложений»**

**Отчет по лабораторной работе №7**

**«Авторизация, работа с формами, Django admin»**

Выполнил:

студент группы ИУ5-53  
Егоров С.А.

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2017 г.

## 1 Задание

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'а 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

## 2 Листинг

### Forms.py:

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

class RegistrationForm(UserCreationForm):
    username = CharField(min_length=5, label='Логин')
    password1 = CharField(min_length=8, widget=PasswordInput,
label='Пароль')
    password2 = CharField(min_length=8, widget=PasswordInput,
label='Повторите ввод')
    email = EmailField(label='Email')
    first_name = CharField(max_length=30, label='Введите имя')
    last_name = CharField(max_length=30, label='Введите фамилию')

    class Meta:
        fields = [
            'username',
            'password1',
            'password2',
            'email',
            'first_name',
            'last_name'
        ]
        model = User

class AuthorizationForm(AuthenticationForm):
    username = CharField(min_length=5, label='Логин')
    password = CharField(min_length=8, widget=PasswordInput,
label='Пароль')
```

### admin.py

```
from labApp.models import *

@admin.register(Users)
class CustomerAdmin(admin.ModelAdmin):
    #fields = ('first_name', 'last_name')
    list_display = ('username', 'full_name', 'email',)
    search_fields = ['last_name', 'first_name']

    def full_name(self, obj):
        return "{} {}".format(obj.last_name, obj.first_name)

    def username(self, obj):
        return "{}".format(obj.user.username)

@admin.register(Departments)
class DepartmentAdmin(admin.ModelAdmin):
    empty_value_display = '-empty-'
    list_display = ('department', 'description',)
    search_fields = ['product_name']

    def department(self, obj):
        return "{}".format(obj.product_name)
```

```

def description(self, obj):
    return "{}".format(obj.description)

```

## views.py

```

from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.http import HttpResponseRedirect
from django.shortcuts import render, redirect
from django.views.generic import *
from labApp.forms import RegistrationForm, AuthorizationForm

class MainPage(TemplateView):
    template_name = 'index.html'

class Registration(FormView):
    template_name = 'registration.html'
    form_class = RegistrationForm
    success_url = '/success/'

    def form_valid(self, form):
        form.save()
        return super(Registration, self).form_valid(form)

class Autorization(FormView):
    template_name = 'autorization.html'
    form_class = AuthorizationForm
    success_url = '/success/'

    def post(self, request, *args, **kwargs):
        form = AuthorizationForm(data= request.POST)
        if form.is_valid():
            user =
authenticate(username=form.cleaned_data['username'],password=form.cleaned_d
ata['password'])
            if user is not None and user.is_active:
                login(request,user)
                return redirect(self.success_url)
            return render(request, self.template_name, {'form':form})

@login_required(login_url='/error/')
def login_success(request):
    return render(request, 'login.html')

def logout_view(request):
    logout(request)
    return HttpResponseRedirect('/error/')

def error_auth(request):
    return render(request, 'logout.html')

```

## models.py

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


class Users(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    email = models.EmailField(max_length=75)
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)

    def __str__(self):
        return "{}".format(self.user)


class Departments(models.Model):
    department_name = models.CharField(max_length=30)
    description = models.CharField(max_length=255, null=True)

    def __str__(self):
        return "{} {}".format(self.department_name, self.description)
```

## urls.py

```
from django.conf.urls import url
from django.contrib import admin
from labApp.views import *

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', MainPage.as_view(), name='main'),
    url(r'^registration/', Registration.as_view(), name='registration'),
    url(r'^authorization/', Autorization.as_view(), name='authorization'),
    url(r'^success/', login_success),
    url(r'^error/', error_auth, name='error_auth'),
    url(r'^logout/', logout_view, name='logout'),
]
```

## base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<meta charset="utf-8">
<head>
    <link rel="stylesheet" type="text/css" href="{% static 'bootstrap.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'bootstrap-theme.css' %}">
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <nav class="navbar navbar-inverse">
        <div class="container">
            <div class="navbar-header">
                <a class="navbar-brand" href="{% url 'main' %}">Lab
            </div>
        </div>
    </nav>
    <div class="container">
```

```

<header>{% block header %}{% endblock
%}</header> <div>
    {% block errors %}{% endblock %}
    {% block body %}{% endblock %}
</div>
</div>
<div class="container">
    <div class="footer">
        <hr>
        <p>сору Егоров Степан 2017</p>
    </div>
</div>
</body>
</html>

```

## registration.html

```

{% extends 'base.html' %}

{% block title %}Регистрация{% endblock %}

{% block header %}<h2 class="text-left">Регистрация</h2>{% endblock %}

{% block body %}
    <form action="{% url 'registration' %}"
        method="POST"> {% csrf_token %}
        {{ form.as_p }}
        <button class="btn-primary"
            type="submit">Зарегистрироваться</button> </form>
{% endblock %}

```

## authorization.html

```

{% extends 'base.html' %}

{% block title %}Авторизация{% endblock %}

{% block header %}<h2>Авторизация</h2>{% endblock %}

{% block body %}
    <form action="{% url 'authorization' %}"
        method="POST"> {% csrf_token %}
        {{ form.as_p }}
        <button class="btn-primary"
            type="submit">Авторизироваться</button> </form>
{% endblock %}

```

## index.html

```

{% extends 'base.html' %}

{% block title %}Главная{% endblock %}

{% block body %}
    <h3><a href="{% url 'registration' %}">Регистрация</a></h3>
    <h3><a href="{% url 'authorization' %}">Авторизация</a></h3>
{% endblock %}

```

## login.html

```

{% extends 'base.html' %}

{% block title %}Успех!{% endblock %}

```

```
{% block header %}<h2>Вы успешно вошли в систему!</h2>{% endblock %}

{% block body %}
    <button type="button" class="btn-default"><a href="{% url 'logout' %}">Выйти</a></button>
{% endblock %}

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

{% block title %}Ошибка!{% endblock %}

{% block header %}<h2>Вы не вошли в систему!</h2>{% endblock %}

{% block body %}
    <button type="button" class="btn-default"><a href="{% url 'registration' %}">Регистрация</a></button>
    <button type="button" class="btn-default"><a href="{% url 'authorization' %}">Авторизация</a></button>
{% endblock %}
```

## Результат

Главная страница:



Регистрация

Авторизация

© Егоров Степан 2017

Авторизация:



Авторизация

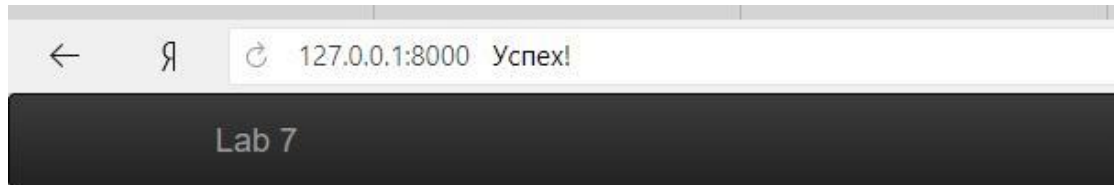
Логин: user1

Пароль: .....

Авторизироваться

© Егоров Степан 2017

Переход на страницу при успешной авторизации\регистрации:

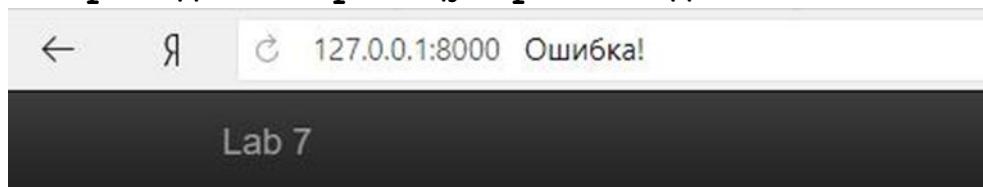


Вы успешно вошли в систему!

[Выйти](#)

© Егоров Степан 2017

Переход на страницу при выходе:



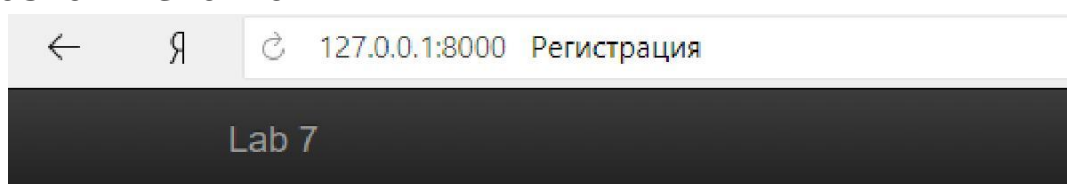
Вы не вошли в систему!

[Регистрация](#)

[Авторизация](#)

© Егоров Степан 2017

Возможные ошибки:



## Регистрация

Логин:

Пароль:



Минимальное количество символов: 8. Длина текста сейчас: 7.

Email:

Введите имя:

Введите фамилию:

[Зарегистрироваться](#)



# Регистрация

- Логин занят

Логин:

Пароль:

Повторите ввод:

Email:

Введите имя:

Введите фамилию:

[Зарегистрироваться](#)

© Егоров Степан 2017

← Я × 127.0.0.1:8000 Регистрация

Lab 7

# Регистрация

Логин:

Пароль:

- Пароли не совпадают

Повторите ввод:

Email:

Введите имя:

Введите фамилию:

[Зарегистрироваться](#)

© Егоров Степан 2017

## Создание суперпользователя:

```
C:\Users\Вкусик\Documents\GitHub\lab_7>manage.py createsuperuser
Username: root
Email address: email.ru
Error: Enter a valid email address.
Email address: email@yandex.ru
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is entirely numeric.
Password:
Password (again):
This password is entirely numeric.
Password:
Password (again):
Superuser created successfully.
```

## Django admin:

### Django administration

#### Site administration

| AUTHENTICATION AND AUTHORIZATION |                       |                          |
|----------------------------------|-----------------------|--------------------------|
| Groups                           | <a href="#">+ Add</a> | <a href="#">✎ Change</a> |
| Users                            | <a href="#">+ Add</a> | <a href="#">✎ Change</a> |
| LABAPP                           |                       |                          |
| Departmentss                     | <a href="#">+ Add</a> | <a href="#">✎ Change</a> |
| Userss                           | <a href="#">+ Add</a> | <a href="#">✎ Change</a> |

#### Recent actions

##### My actions

None available