

**Отчет по лабораторной работе № 7**  
**по курсу**  
**"Разработка интернет приложений"**

**" Работа с формами,авторизация,django-admin "**

Бодунов А.Г., ИУ5-53

Москва, МГТУ - 2016 год

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 вы освоите инструменты Django по работе с авторизацией и реализуете простейшую авторизацию. Напоследок, вы познакомитесь 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`'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

Модуль `Views.py`

```

def login_hand(request):
    args={}
    args.update(csrf(request))
    if request.POST:
        username=request.POST['username']
        password=request.POST['password']
        user =auth.authenticate(username=username,password=password)
        if user is not None:
            auth.login(request,user)
            args['login_error']=''
            return HttpResponseRedirect('/')
        else:
            args['login_error']='Пользователь не найден'
            return render_to_response('hand_login.html',args)
    else:
        return render_to_response('hand_login.html',args)

def login(request):
    redirect_url = '/'
    if request.method == 'POST':
        redirect_url = '/'
        form = LoginForm(request.POST)
        if form.is_valid():
            user = auth.authenticate(username=form.cleaned_data['login'],
                                    password=form.cleaned_data['password'])
            if user is not None:
                auth.login(request, user)
                return HttpResponseRedirect(redirect_url)
            else:
                form.add_error(None, 'invalid login/password')
        else:
            form = LoginForm()
    return render(request, 'login.html', {'form':form, 'continue': redirect_url})

def signup(request):
    if request.method == 'POST':
        form = SignupForm(request.POST)
        if form.is_valid():
            user = form.save()
            return HttpResponseRedirect('/login/')
    else:
        form = SignupForm()
    return render(request, 'signup.html', {'form': form,
                                          'type': 'Registration'})

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

class InitView(View):
    def get(self,request):
        return render(request,'init.html')

```

[illegible]

```
forms.py x admin.py x lab7_777\urls.py x app1\urls.py x
__author__ = 'Work'

from django.conf.urls import url
from django.contrib import admin
from django.contrib.auth import views
from . import views

urlpatterns=[
    url(r'^admin/', admin.site.urls),
    url(r'^signup/', views.signup, name='signup'),
    url(r'^logout/', views.logout_view, name='logout'),
    url(r'^login/', views.login_hand, name='login'),
    ##url(r'', views.logout_view, name='index'),
    url(r'',views.InitView.as_view(), name='init')
]
```

```
forms.py x admin.py x lab7_777\urls.py x app1\urls.py x views.py x hand_login.html x
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <form action="login" method="post">
    {% csrf_token %}
    <label for="username">имя пользователя</label>
    <input type="text" name="username" id="username"><p></p>
    <label for="password">пароль</label>
    <input type="text" name="password" id="password"><p></p>
    <input class="button" type="submit" >
    <p>{{login_error}}</p>
  </form>
</body>
</html>
```

```
forms.py x admin.py x lab7_777.urls.py x appl\urls.py x views.py x hand_login.html x logout.html x signup.htm

<html>
<head>
{% load static %}
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>
<body>

{{ form.non_field_errors }}
<form action="/signup/" method="post" class="form-horizontal" enctype="multipart/form-data">
    {% csrf_token %}
    {% for i in form %}
    <div class="form-group">
        <label class="col-sm-4">{{ i.label }}</label>
        <div class="col-sm-8">
            <div>
                {{ i }}
            </div>
            <div>
                {{ i.errors }}
            </div>
        </div>
    </div>
    {% endfor %}

    <div class="form-group">
        <div class="col-sm-offset-4 col-sm-8">
            <input type="submit" value="Зарегистрировать меня" class="btn btn-default"/>
        </div>
    </div>
</form>
</body>
</html>
```

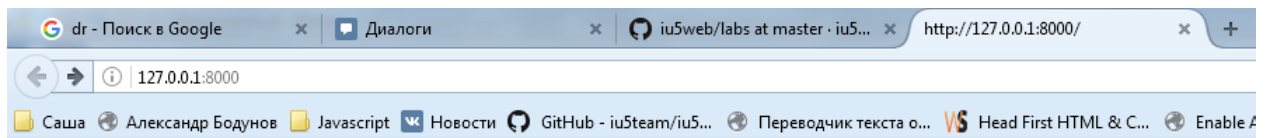
```
forms.py x admin.py x lab7_777.urls.py x appl\urls.py x views.py x hand_login.html x logout.html x signup.html x login.html

html body
<html>
<head>
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>
<body>

{{ form.non_field_errors }}
<form action="/login/" method="post" class="form-horizontal">
    {% csrf_token %}
    {% for i in form %}
    <div class="form-group">
        <label class="col-sm-2">{{ i.label }}</label>
        <div class="col-sm-10">
            <div>
                {{ i }}
            </div>
        </div>
    </div>
    {% endfor %}

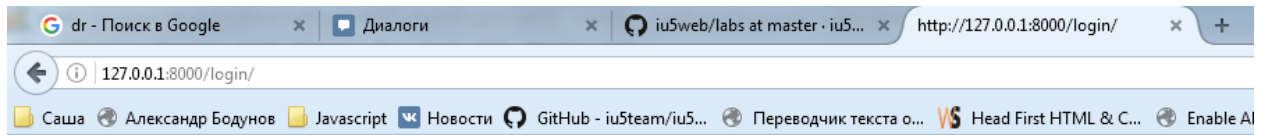
    <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
            <input type="submit" value="Войти" class="btn btn-default"/>
        </div>
    </div>
</form>
<a href="{% url 'init' %}">На главную</a>
</body>
</html>
```

Результаты работы программы:



Здравствуйте, Гость

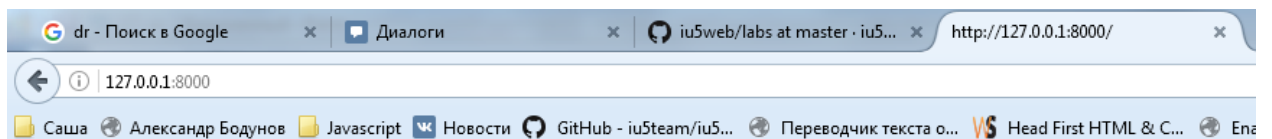
[Зарегистрироваться](#) [Войти](#)



имя пользователя admin

пароль

Отправить запрос



Здравствуйте, admin

[Выйти](#)

Логин	<input type="text" value="ff"/> <ul style="list-style-type: none"><li>Ensure this value has at least 5 characters (it has 2).</li></ul>
Пароль	<input type="password"/> <ul style="list-style-type: none"><li>Ensure this value has at least 8 characters (it has 1).</li></ul>
Повторите пароль	<input type="password"/>
Адрес электронной Почты	<input type="text" value="h"/> <ul style="list-style-type: none"><li>Enter a valid email address.</li></ul>
Имя	<input type="text" value="g"/>
Фамилия	<input type="text" value="g"/>
	<input type="button" value="Зарегистрировать меня"/>