

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ Н.Э. БАУМАНА

Кафедра ИУ-5  
«Системы обработки информации и управления»

Отчет по лабораторной работе № 7  
По предмету «Разработка интернет-приложений»

Выполнил: Алтунин Н.С.  
Группа: ИУ5-53

## **1. Задание**

**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'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

## Страница `urls.py`:

```
from django.conf.urls import url
from django.contrib import admin
from lab6.views import *
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^books$', main, name='main'),
    url(r'^signup$', signup, name='signup'),
    url(r'^signin$', signin, name='signin'),
    url(r'^logout$', logout, name='logout'),
    url(r'^success$', success, name='success'),
    url(r'^booksModel$', booksModel, name='booksModel'),
]
```

## Файл signup.html:

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Sign up</button>
  </form>
</body>
</html>
```

## Страница signin.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  {{ form }}
  <form method="POST">
    {% csrf_token %}
    <p>
      <label type="text" name="username"><input type="text"
name="username"></label>
    </p>
    <p>
      <label type="text" name="password"><input type="text"
name="password"></label>
    </p>
    <button type="submit">Sign in</button>
  </form>
</body>
</html>
```

## Файл models.py:

```
from django.db import models
```

```
from django.contrib import admin
from django.contrib.auth.models import User
# Create your models here.
class Book(models.Model):
    name = models.CharField(max_length=45)
    author = models.CharField(max_length=45)
    description = models.CharField(max_length=225)
```

```

class Meta:
    db_table = 'Books'
@admin.register(Book)
class BookAdmin(admin.ModelAdmin):
    fields = ('name', 'author', 'description', 'NEW')
    list_display = ('name', 'author', 'description', 'NEW')
    search_fields = ('name', 'author')
    def NEW(self, obj):
        return 'new'
class signupModel(models.Model):
    name = models.CharField(max_length=45)
    author = models.CharField(max_length=45)
    description = models.CharField(max_length=225)

```

## Файл views.py:

```

from django.shortcuts import render
from django.views import View
from django.contrib import auth
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.contrib.auth.decorators import login_required
from lab6.models import Book
from lab6.forms import *
# Create your views here.
def main(request):
    books = Book.objects.all()
    return render(request, 'main.html', {'books': books} )
def signup(request):
    if request.user.is_authenticated():
        return HttpResponseRedirect('/')
    if request.method == "POST":
        form = SignupForm(request.POST)
        if form.is_valid():
            user = form.save()
            auth.login(request, user)
            return HttpResponseRedirect('/')
    else:
        form = SignupForm()
    return render(request, 'signup.html', {'form': form})
def signin(request):
    redirect = '/success'
    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(redirect)
    else:
        return HttpResponseRedirect('/')
    return render (request, 'signin.html',)
@login_required()
def success(request):
    if request.user.is_authenticated():
        return render(request, 'success.html')
def logout(request):
    auth.logout(request)
    return HttpResponseRedirect('/')
def booksModel(request):

```

```
return render(request, 'formmodel.html', {'form': ArticleForm()})
```

**Скриншоты:**

***admin***

Select book to change

[ADD BOOK](#) + Action:  0 of 3 selected

<input type="checkbox"/>	NAME	AUTHOR	DESCRIPTION	NEW
<input type="checkbox"/>	<a href="#">Телекоммуникации и Сети</a>	Галкин	Учебное пособие по сетям	new
<input type="checkbox"/>	<a href="#">Властелин колец</a>	Толкиен	Трилогия	new
<input type="checkbox"/>	<a href="#">Принц Амбера</a>	Роджер Желязны	-	new

3 books

## signup

Login: Firstname: Lastname: E-mail: Password: Repeat password: