

Отчет по лабораторной работе №7
«Авторизация, регистрация»
по курсу «Разработка интернет
приложений»

Выполнил студент ИУ5-51 Никитин К.И.

2017 год

| Файл login.html

```
{% extends "base.html" %}
{%block title%}Вход{% endblock %}
{% block login %}
<div>
  <form action="/auth/login/" method="post" class="form-horizontal" >
    {% csrf_token %}
    <label for="username">Имя пользователя</label>
    <input type="text" name="username" id="username">
    <label for="password">Пароль</label>
    <input type="text" name="password" id="password">
    <input class="button" type="submit" value="Войти">
  </form>
  {% if login_error %}
    <label class="errors">{{ login_error }}</label>
  {% endif %}
  <div><a href="/auth/register/">Регистрация через UserRegisterForm</a></div>
  <div class="col-sm-12"><a href="/auth/registration/">Регистрация через
RegistrationForm</a></div>
</div>
{% endblock %}
```

файл base.html

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Booking site">
    <meta name="author" content="Никитин Кирилл">
    {% load static %}
    <title>home</title>
    <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static 'css/ie10-viewport-bug-workaround.css' %}"
rel="stylesheet">
    <link href="{% static 'css/styles.css' %}" rel="stylesheet">
    <script src="{% static 'js/ie-emulation-modes-warning.js' %}"></script>
  </head>
  <body>
    <div>
      {% if username %}
        <a href="/auth/logout/">Выйти ({{ username }})</a>
        <p>Hello, World!!</p>
      {% else %}
        <a href="/auth/login/">Войти</a>
      {% endif %}
    </div>
    {% block register %}{% endblock %}
    {% block login %}{% endblock %}
  </body>
</html>
```

файл register.html

```
{% extends "base.html" %}
{%block title%}Регистрация{% endblock %}
```

```
{% block register %}
    <div>
        <form action="/auth/registration/" method = 'post'>
            {% csrf_token %}
            {{ form }}
            <input class="button" type="submit" value="Зарегистрироваться">
        </form>
    </div>
{% endblock %}
```

файл register0.html

```
{% extends "base.html" %}
{%block title%}Регистрация{% endblock %}
{% block register %}
    <div>
        <form action="/auth/register/" method = 'post'>
            {% csrf_token %}
            {{ form }}
            <input class="button" type="submit" value="Зарегистрироваться">
        </form>
    </div>
{% endblock %}
```

файл views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render, redirect
from django.contrib import auth
from django.contrib.auth.forms import UserCreationForm
from .forms import *
from django.views.decorators import csrf
# Create your views here.
def login(request):
    args = {}
    if request.POST:
        username = request.POST.get('username', '')
        password = request.POST.get('password', '')
        user = auth.authenticate(username=username, password=password)
        #print(auth.authenticate(username=username, password=password))
        if user is not None:
            auth.login(request, user)
            return redirect("/")
        else:
            args['login_error'] = "Пользователь не найден"
            return render(request, 'login.html', args)
    else:
        return render(request, 'login.html', args)
def logout(request):
    auth.logout(request)
    return redirect("/")
def register(request):
    args = {}
    args['form'] = UserCreationForm()
    if request.POST:
        newuser_form = UserCreationForm(request.POST)
        if newuser_form.is_valid():
            newuser_form.save()
```

```

        newuser = auth.authenticate(username=request.POST["username"],
password=request.POST["password1"])
        auth.login(request,newuser)
        return redirect('/')
    else:
        args['form'] = newuser_form
    return render(request, "register0.html", args)
def registration(request):
    context = {}
    if request.method == 'POST':
        form = RegistrationForm(request.POST, request.FILES)
        is_val = form.is_valid()
        if is_val:
            data = form.cleaned_data
            if data['password'] != data['password2']:
                is_val = False
                form.add_error('password2', ['Пароли не совпадают'])
            if models.User.objects.filter(username=data['username']).exists():
                form.add_error('username', ['Пользователь с данным логином уже
существует'])
                is_val = False
            if models.User.objects.filter(email=data['email']).exists():
                form.add_error('email', ['Пользователь с данной электронной
почтой уже зарегистрирован'])
                is_val = False
            if is_val:
                customer = form.save(commit=False)
                customer.user = models.User.objects.create_user(data['username'],
data['email'], data['password'])
                customer.save()
                return redirect('/')
        else:
            form = RegistrationForm()
            context['form'] = form
            return render(request, "register.html", context)

```

файл models.py

```

from django.db import models
from django.contrib.auth.models import User
# Create your models here.
class Customer(models.Model):
    class Meta:
        db_table = "Customer"
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    username = models.CharField(max_length=30)
    first_name = models.CharField(max_length=30)
    second_name = models.CharField(max_length=30)
    gender = models.CharField(max_length=1)
    password = models.CharField(max_length=30)
    password2 = models.CharField(max_length=30, default='1234')
    email = models.EmailField(max_length=45, default="home@mail.ru")
    photo = models.FileField(upload_to='photo/', blank=True,
default='photo/default.png')

```

файл forms.py

```

from django import forms
from lab7 import models
class RegistrationForm(forms.ModelForm):
    username = forms.CharField(min_length=5, label='Логин')
    password = forms.CharField(min_length=8, widget=forms.PasswordInput,
label='Пароль')
    password2 = forms.CharField(min_length=8, widget=forms.PasswordInput,
label='Повторите ввод')
    email = forms.EmailField(label='Email')
    last_name = forms.CharField(label='Фамилия')
    first_name = forms.CharField(label='Имя')
    gender = forms.CharField(label='Пол')
    photo = forms.FileField(label='Аватар',
widget=forms.ClearableFileInput(attrs={'class': 'ask-signup-avatar-
input'}), required=False)
    class Meta:
        model = models.Customer
        fields = ('username', 'password', 'password2', 'email', 'last_name',
'first_name', 'gender', 'photo')

```

Результаты.

Вход - Mozilla Firefox

127.0.0.1:8000/auth/login/

Войти

Имя пользователя Пароль Войти

Регистрация через UserRegisterForm

Регистрация через RegistionForm

Регистрация - Mozilla Firefox

127.0.0.1:8000/auth/register/

Войти

Имя пользователя:

Обязательное поле. Не более 150 символов. Только буквы, цифры и символы @/./+/_ . Пароль:

- Ваш пароль не должен совпадать с вашим именем или другой персональной информацией или быть слишком похожим на неё.
- Ваш пароль должен содержать как минимум 8 символов.
- Ваш пароль не может быть одним из широко распространённых паролей.
- Ваш пароль не может состоять только из цифр.

Подтверждение пароля:

Для подтверждения введите, пожалуйста, пароль ещё раз. Зарегистрироваться

Регистрация - Mozilla Firefox

Диалоги Unicorn! - GitHub Регистрация +

127.0.0.1:8000/auth/registration/ 200% Поиск

Войти

Логин: Пароль: Повторите ввод:

Email: Фамилия:

Имя: Пол:

Аватар:

Обзор... Файл не выбран.

Зарегистрироваться

Выберите customer для изменения | Административный сайт Django - Mozilla Firefox

Диалоги Unicorn! - GitHub Выберите customer для и... +

127.0.0.1:8000/admin/lab7/customer/ 130% Поиск

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ROOT. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало · Lab7 · Customers

Выберите customer для изменения ДОБАВИТЬ CUSTOMER +

Действие: Выполнить Выбрано 0 объектов из 1

☐ CUSTOMER

☐ Customer object (1)

1 customer