Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ (ИУ)

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)

О Т Ч Е Т

по лабораторной работе № 6

по дисциплине: Разработка интернет-приложений

на тему: Работа с формами, авторизация и модуль администрирования в Django

Студент ИУ5-53 _____ **А.С. Волков**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

2019 г.

1. 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 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.

2. Исходные коды

2.1. Lab_6/urls.py

```
from django.contrib import admin
from django.urls import path
from django.conf.urls import url

from user import views
from user.views import EventView, UserView, index

urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^events/', EventView.as_view()),
    url(r'^users/', UserView.as_view()),
    url(r'^registration/', views.registration, name='registration'),
    url(r'^login/', views.login, name='login'),
    url(r'^logout/', views.logout, name='logout'),
    path('', index),
]
```

2.2. 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
from django.views import View
from user.models import *
from .forms import RegistrationForm, LoginForm
from django.contrib.auth import authenticate


def index(request):
    return render(request, 'base.html')


class EventView(View):
    def get(self, request):
        events = Event.objects.all()
        users = User.objects.all()
        data = {
            'events': events,
            'users': users
        }
```

```
        return render(request, 'events.html', data)


class UserView(View):
    def get(self, request):
        users = User.objects.all()
        data = {
            'users': users
        }
        return render(request, 'users.html', data)


def registration(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        password = request.POST.get('password')
        password_repeat = request.POST.get('password2')
        result =
Customers.objects.filter(username=request.POST.get('username')).values()
        list_result = [entry for entry in result]
        if form.is_valid() & (password == password_repeat) & (list_result == []):
            params = {
                'username': request.POST.get("username"),
                'password': request.POST.get("password"),
                'email': request.POST.get("email"),
                'first_name': request.POST.get("first_name"),
                'last_name': request.POST.get("last_name")
            }
            customer = Customers.objects.create(**params)
            customer.save()
            return HttpResponseRedirect('/login/')
    else:
        form = RegistrationForm()

    return render(request, 'registration.html', {'form': form})


@login_required
def login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        result = Customers.objects.filter(username=request.POST.get('username'),

password=request.POST.get('password')).values()
        list_result = [entry for entry in result]
        if form.is_valid() & (list_result != []):
            return HttpResponseRedirect('/logout/')
    else:
        form = LoginForm()

    return render(request, 'login.html', {'form': form})


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

2.3. forms.py

```
from django import forms
```

```
class RegistrationForm(forms.Form):
    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.CharField(min_length=7, label='Email')
    first_name = forms.CharField(min_length=2, label='Имя')
    last_name = forms.CharField(min_length=2, label='Фамилия')


class LoginForm(forms.Form):
    username = forms.CharField(min_length=5, label='Логин')
    password = forms.CharField(min_length=8, widget=forms.PasswordInput,
label='Пароль')
```

2.4. admin.py

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


class EventAdmin(admin.ModelAdmin):
    fieldsets = [
        (None, {'fields': ['name_event']}),
        ('Date information', {'fields': ['date_event']}),
    ]
    list_display = ('name_event', 'date_event')
    list_filter = ['date_event']
    search_fields = ['name_event']


admin.site.register(Event, EventAdmin)
```

2.5. models.py

```
from django.db import models


class Event(models.Model):
    name_event = models.CharField(max_length=100)
    date_event = models.DateTimeField('date realized')

    def __str__(self):
        return self.name_event


class User(models.Model):
    event = models.ForeignKey(Event, on_delete=models.CASCADE)
    name_user = models.CharField(max_length=30)
    age_user = models.IntegerField(default=0)
    field = models.ImageField(blank=True, null=True, upload_to='./static/photo'),

    def __str__(self):
        return self.name_user


class Customers(models.Model):
    username = models.CharField(max_length=10)
```

```
    password = models.CharField(max_length=15)
    email = models.CharField(max_length=20)
    last_name = models.CharField(max_length=10)
    first_name = models.CharField(max_length=10)
```

2.6. login.html

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

2.7. logout.html

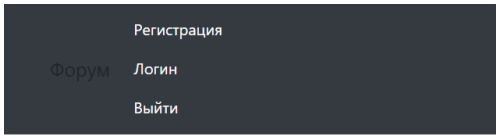
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<b>Вы вошли в акаунт.</b><br>
<a href="/login">Выйти из акаунта</a>
</body>
</html>
```

2.8. registration.html

```
{% extends 'base.html' %}
{% block title %}Регистрация{% endblock %}
{% block body %}
<form method="POST">
   {% csrf_token %}
   {{ form.as_p }}
   <button type="submit">Зарегистрировать</button>
</form>
{% endblock %}
```

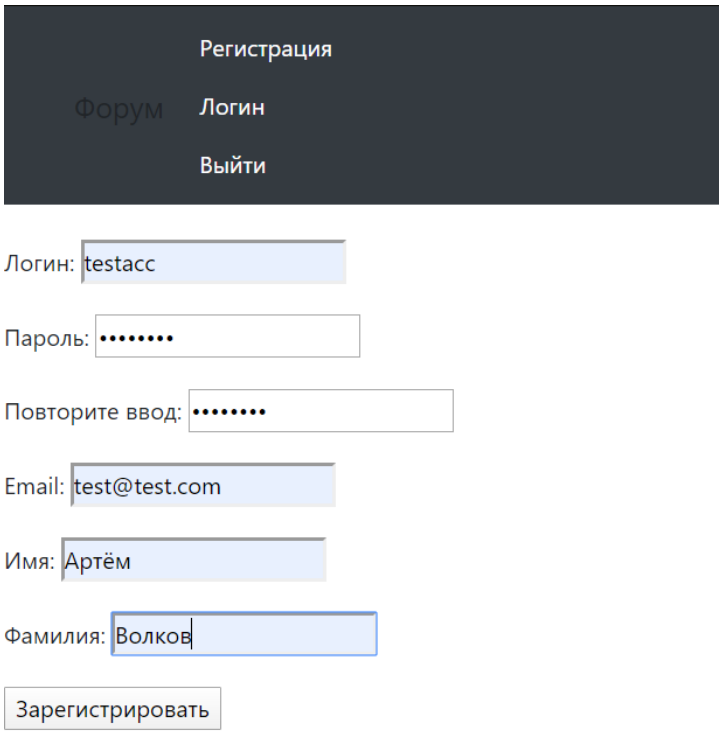
3. Скриншоты

3.1. Главная страница

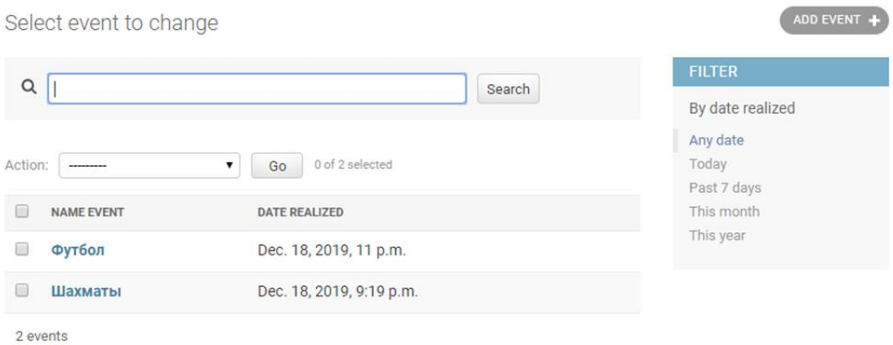


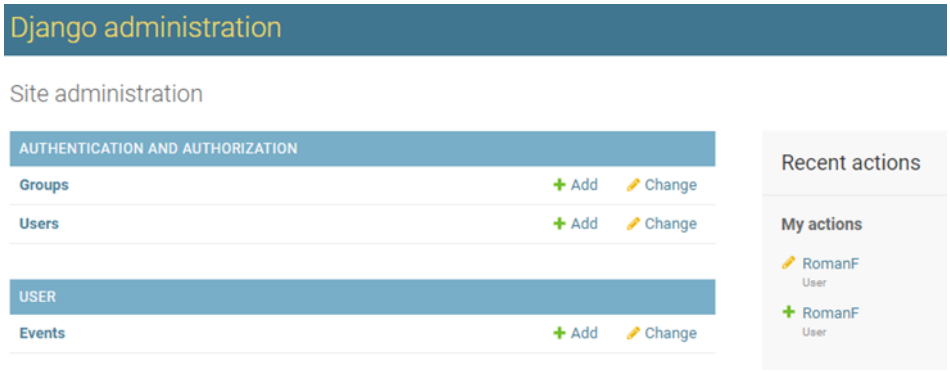
Главная страница

3.2. Регистрация



3.3. Панель администратора





3.4. Создание суперпользователя

```
C:\Users\Артём\Google Диск\Учёба\5 сем\РИП\Lab6_final>python manage.py createsuperuser
Username: superuser
Email address: super@user.com
Password:
Password (again):
Superuser created successfully.
```