

Отчет по ДЗ
Курс «Разработка Интернет-приложений»

Выполнил:

студент группы ИУ5-54

(подпись)

Харлашкин А. И.

"__" _____ 2017 г.

Проверил:

Преподаватель каф. ИУ5

(подпись)

Гапанюк Ю. Е.

"__" _____ 2017 г.

Москва, МГТУ – 2017 г.

Описание задания лабораторной работы

Техническое задание на домашнее задание по курсу РИП

Задание

Разработать веб-сервис на базе технологий: Python, Django, JS, MySQL.

Предметные области

23	Пользователь	Ставка	Боксёр
----	--------------	--------	--------

Листинг программы

Apps.py

```
from django.apps import AppConfig
```

```
class AdminConfig(AppConfig):  
    name = 'users'
```

connections.py

```
try:  
    import pymysql  
    pymysql.install_as_MySQLdb()  
except ImportError:  
    pass
```

settings.py

```
"""  
Django settings for lab6 project.  
  
Generated by 'django-admin startproject' using Django 1.10.3.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/1.10/topics/settings/  
  
For the full list of settings and their values, see  
https://docs.djangoproject.com/en/1.10/ref/settings/  
"""  
  
import os  
  
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)  
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.10/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '&14-rmry%jq3w%d)78)rud8uiki=l8m=ycn%c)1k0d126d=ncd'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'users',
]

MIDDLEWARE = [
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'lab6.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'lab6.wsgi.application'

# Database
# https://docs.djangoproject.com/en/1.10/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'first_db',
        'USER': 'admin',
        'PASSWORD': '12345',
    }
}

```

```

        'HOST': '127.0.0.1',
        'PORT': '3306',
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.10/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
        'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/1.10/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.10/howto/static-files/

STATIC_URL = '/static/'
MEDIA_URL = '/Images/'
MEDIA_ROOT = 'C:/Users/Толя/Desktop/РМП/JP_№6/lab6-master/Images'

```

Urls.py

```

from django.conf.urls import url
from django.contrib import admin
from users.views import *
from django.conf import settings
from django.conf.urls.static import static

```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^fights/$', listing, name='fights'),
    url(r'^registration/$', registration, name='registration'),
    url(r'^login/$', login1, name='login'),
    url(r'^fight/(?P<id>\d+)$', FightView.as_view(), name='fight_url'),
    url(r'^logout$', Logout.as_view(), name='logout'),
    url(r'^newFight$', new_fight, name='NewFight'),
    url(r'^addBill$', add_bill, name='add_bill'),
    url(r'^makeBet/(?P<id>\d+)$', make_bet, name='make_bet'),
    url(r'^create/$', create_boxer, name='create_boxer'),
    url(r'^fights2/$', Online.as_view()),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Admin.py

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

admin.site.register(User)
admin.site.register(Bet)
admin.site.register(Bill)
admin.site.register(Boxer)
admin.site.register(Fight)
```

Apps.py

```
from django.apps import AppConfig

class UsersConfig(AppConfig):
    name = 'users'
```

forms.py

```
from django import forms

class CreateForm(forms.Form):
    idBoxer = forms.IntegerField(label='idBoxer', required=False)
    title = forms.CharField(label='title', required=False)
    boxer_first_name = forms.CharField(label='boxer_first_name',
    required=False)
    boxer_last_name = forms.CharField(label='boxer_last_name',
    required=False)
```

Models.py

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

class User(models.Model):    # пользователь
    username = models.CharField(max_length=255, unique=True)
    email = models.EmailField(max_length=255, null=False)
```

```

bill = models.IntegerField(null=True)
name = models.CharField(max_length=255, null=False)
surname = models.CharField(max_length=255, null=False)

class Bet(models.Model): # ставка
    # idBet = models.IntegerField(unique=True)
    userLogin2 = models.CharField(max_length=255, null=True)
    size = models.IntegerField(null=False)
    result = models.BooleanField(null=False)
    idFight1 = models.IntegerField(null=True)

class Bill(models.Model): # счет
    userLogin1 = models.CharField(max_length=255, null=True)
    bill = models.IntegerField(null=True)

class Fight(models.Model): # бой
    idFight = models.IntegerField(unique=True)
    date = models.DateField(null=False)
    time = models.TimeField(null=False)
    boxer1 = models.TextField(max_length=255, null=False)
    boxer2 = models.TextField(max_length=255, null=False)
    idBoxer1 = models.IntegerField(null=True)
    idBoxer2 = models.IntegerField(null=True)
    #img = models.ImageField(upload_to='/Images', height_field=90,
width_field=150, null=True)

class Boxer(models.Model): # Боксёр
    idBoxer = models.IntegerField(unique=True)
    title = models.TextField(max_length=255) # весовая категория
    boxer_first_name = models.CharField(max_length=255, null=False)
    boxer_last_name = models.CharField(max_length=255, null=False)
    fights = models.ManyToManyField(Fight)

# -----

class RegisterForm(forms.Form):
    login = forms.CharField(label='Login', min_length=5)
    password = forms.CharField(label='Password', min_length=8,
widget=forms.PasswordInput)
    password2 = forms.CharField(label='Password', min_length=8,
widget=forms.PasswordInput)
    email = forms.CharField(label='Email', min_length=1)
    # userBill = forms.IntegerField(label='Bill')
    surname = forms.CharField(label='Surname', min_length=1)
    name = forms.CharField(label='Name', min_length=1)

    def clean(self):
        cleaned_data = super(RegisterForm, self).clean()
        password = cleaned_data.get('password')
        password2 = cleaned_data.get('password2')
        if password != password2:
            raise forms.ValidationError("Пароли не совпадают")
        usrs = User.objects.filter(username=cleaned_data.get('login'))
        if len(usrs) > 0:
            raise forms.ValidationError("Пользователь с данным логином уже
существует")

```

views.py

```
from users.models import *
from django.views.generic import View, ListView
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from django.contrib.auth import authenticate, logout
from django.contrib import auth
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
from django.contrib.auth.models import User
from django.contrib.auth.hashers import make_password
from users.forms import CreateForm

class CourseListView(ListView):
    template_name = "fights_list.html"
    model = Fight
    context_object_name = "fights2"
    paginate_by = 10

class Online(CourseListView):
    template_name = 'fights2.html'

def listing(request):
    fights = Fight.objects.all()
    # paginator = Paginator(fights_list, 5)
    bills = Bill.objects.filter(userLogin1=request.user.username)
    '''page = request.GET.get('page')
    try:
        fights = paginator.page(page)
    except PageNotAnInteger:
        # If page is not an integer, deliver first page.
        fights = paginator.page(1)
    except EmptyPage:
        # If page is out of range (e.g. 9999), deliver last page of results.
        fights = paginator.page(paginator.num_pages)'''
    return render(request, 'index.html', {"fights": fights, "bills": bills})

def registration(request):
    errors = []
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors.append('Введите логин')

        if len(str(username)) < 5: # !!!!!!!!!!!!!!!!!!!!!
            errors.append('Логин должен быть не меньше 5 символов')

        password = request.POST.get('password')
        if len(str(password)) < 6: # !!!!!!!!!!!!!!!!!!!!!
            errors.append('Длина пароля должна превышать 6 символов')

        password2 = request.POST.get('password2')
        if password != password2:
            errors.append('Пароли должны совпадать')

        e_mail = request.POST.get('E_mail')
        surname = request.POST.get('surname')
        name = request.POST.get('name')
        if (not username) or (not password) or (not password2) or (not
e_mail) or (not surname) or (not name):
```

```

        errors.append('Все поля должны быть заполнены')
    else:
        if len(errors) == 0:
            users = User.objects.filter(username=username)
            if len(users) != 0:
                errors.append('Пользователь с таким логином уже
существует')
            return render(request, 'registration.html', {'errors':
errors, 'username': username,
                                                         'E_mail':
e_mail, 'surname': surname, 'name': name})
        else:
            u = User()
            u.username = username
            u.password = make_password(password)
            u.email = e_mail
            u.surname = surname
            u.name = name
            u.is_staff = False
            u.is_active = True
            u.is_superuser = False
            u.save()
            b = Bill()
            b.userLogin1 = username
            b.bill = 0
            b.save()
            return HttpResponseRedirect('/login/')

    return render(request, 'registration.html', {'errors': errors,
'username': username,
                                                         'E_mail': e_mail,
'surname': surname, 'name': name})

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

def login1(request):
    errors = []
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors.append('Введите логин')

        password = request.POST.get('password')
        if not password:
            errors.append('Введите пароль')

        user = authenticate(username=username, password=password)

        if user:
            auth.login(request, user)
            return redirect('/fights/')
        else:
            errors.append('Неправильный логин или пароль')
    return render(request, 'login.html', {'errors': errors, 'username':
username})

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

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

    def get_context_data(self, **kwargs):

```



```

    Fights = Fight.objects.all()
    context = dict(Fights=Fights)
    return context'''

class FightView(View):
    def get(self, request, id):
        fight = Fight.objects.filter(idFight=int(id))
        boxer = Boxer.objects.all()
        bills = Bill.objects.filter(userLogin1=request.user.username)
        logs1 = Bet.objects.filter(idFight1=int(id))
        return render(request, 'fight.html', {'fight': fight, 'boxer': boxer,
"bills": bills, "logs1": logs1})

class Logout(View):
    success_url = "/"

    def get(self, request):
        logout(request)
        return HttpResponseRedirect("fights/")

def new_fight(request):
    errors = []
    if request.method == 'POST':
        id_fight = request.POST.get('idFight')
        date = request.POST.get('date')
        time = request.POST.get('time')
        boxer1 = request.POST.get('boxer1')
        boxer2 = request.POST.get('boxer2')
        id_boxer1 = request.POST.get('idBoxer1')
        id_boxer2 = request.POST.get('idBoxer2')
        if (not id_fight) or (not date) or (not time) or (not boxer1) or (not
boxer2) \
            or (not id_boxer1) or (not id_boxer2):
            errors.append('Все поля должны быть заполнены')
            return render(request, 'newFight.html', {'errors': errors,
'idFight': id_fight, 'date': date,
'boxer1': boxer1, 'boxer2': boxer2,
'idBoxer1': id_boxer1,
'idBoxer2': id_boxer2})
        else:
            if len(errors) == 0:
                id_fight1 = Fight.objects.filter(idFight=id_fight)
                id_boxer1_1 = Boxer.objects.filter(idBoxer=id_boxer1)
                id_boxer2_1 = Boxer.objects.filter(idBoxer=id_boxer2)
                if len(id_fight1) != 0:
                    errors.append('Бой с таким id уже существует')
                    return render(request, 'newFight.html', {'errors':
errors, 'idFight': id_fight, 'date': date,
'boxer1': boxer1, 'boxer2': boxer2,
'idBoxer1':
id_boxer1, 'idBoxer2': id_boxer2})
                elif len(id_boxer1_1) == 0:
                    errors.append('Боксер1 с таким id не существует')
                    return render(request, 'newFight.html', {'errors':
errors, 'idFight': id_fight, 'date': date,
'boxer1': boxer1, 'boxer2': boxer2,
'idBoxer1':
id_boxer1, 'idBoxer2': id_boxer2})

```

```

        elif len(id_boxer2_1) == 0:
            errors.append('Боксер2 с таким id не существует')
            return render(request, 'newFight.html', {'errors':
errors, 'idFight': id_fight, 'date': date,
'boxer1': boxer1, 'boxer2': boxer2,
'idBoxer1':
id_boxer1, 'idBoxer2': id_boxer2})
        fight = Fight()
        fight.idFight = id_fight
        fight.date = date
        fight.time = time
        fight.boxer1 = boxer1
        fight.boxer2 = boxer2
        fight.idBoxer1 = id_boxer1
        fight.idBoxer2 = id_boxer2
        fight.save()
        return HttpResponseRedirect('/fight/'+str(id_fight))
    return render(request, 'newFight.html', {'errors': errors})

def add_bill(request):
    errors = []
    if request.method == 'POST':
        if request.user.is_authenticated:
            bank_card = request.POST.get('bank_card')
            cvv_cvc_code = request.POST.get('cvv_cvc_code')
            sym = request.POST.get('sym')
            if (not bank_card) or (not cvv_cvc_code) or (not sym):
                errors.append('Все поля должны быть заполнены')
                return render(request, 'addBill.html', {'errors': errors,
'bank_card': bank_card,
'cvv_cvc_code':
cvv_cvc_code, 'sym': sym})
            if int(bank_card) <= 0:
                errors.append('Введите положительный номер банковской карты')
                return render(request, 'addBill.html', {'errors': errors,
'bank_card': bank_card,
'cvv_cvc_code':
cvv_cvc_code, 'sym': sym})
            if int(cvv_cvc_code) <= 0:
                errors.append('Введите положительный CVV/CVC код')
                return render(request, 'addBill.html', {'errors': errors,
'bank_card': bank_card,
'cvv_cvc_code':
cvv_cvc_code, 'sym': sym})
            if len(errors) == 0:
                bs = Bill.objects.filter(userLogin1=request.user.username)
                for b in bs:
                    b.bill += int(sym)
                    b.save()
                return HttpResponseRedirect('/fights/')
            else:
                errors.append('Авторизуйтесь')
                return render(request, 'addBill.html', {'errors': errors})
    return render(request, 'addBill.html', {'errors': errors})

def make_bet(request, id):
    fight = Fight.objects.filter(idFight=int(id))
    box1 = fight[0]
    errors = []
    if request.method == 'POST':
        if request.user.is_authenticated:

```

```

        size_bet = request.POST.get('size_bet')
        result1 = request.POST.get('boxer')
        if (not size_bet) or (not result1):
            errors.append('Все поля должны быть заполнены')
            return render(request, 'makeBet.html', {'errors': errors,
'box1': box1, 'size_bet': size_bet})
        if int(size_bet) <= 0:
            errors.append('Ставка должна быть положительным числом')
            return render(request, 'makeBet.html', {'errors': errors,
'box1': box1, 'size_bet': size_bet})
        if len(errors) == 0:
            bills = Bill.objects.filter(userLogin=request.user.username)
            for bil in bills: # цикл выполнится один раз
                if (int(bil.bill) - int(size_bet)) >= 0:
                    bet = Bet()
                    bet.userLogin2 = request.user.username
                    bet.size = size_bet
                    bet.idFight1 = id
                    if result1 == 'boxer1':
                        bet.result = True
                    else:
                        bet.result = False
                    bet.save()
                    bil.bill -= int(size_bet)
                    bil.save()
                else:
                    errors.append('Недостаточно средств на счету для
ставки')
            return render(request, 'makeBet.html', {'errors':
errors, 'box1': box1, 'size_bet': size_bet})
            return HttpResponseRedirect('/fights/')
        else:
            errors.append('Авторизуйтесь')
            return render(request, 'makeBet.html', {'errors': errors, 'box1':
box1})
        return render(request, 'makeBet.html', {'errors': errors, 'box1': box1})

def create_boxer(request):
    if request.method == 'POST':
        form = CreateForm(request.POST)
        if form.is_valid():
            boxer = Boxer()
            boxer.idBoxer = request.POST.get('idBoxer')
            boxer.title = request.POST.get('title')
            boxer.boxer_first_name = request.POST.get('boxer_first_name')
            boxer.boxer_last_name = request.POST.get('boxer_last_name')
            boxer.save()
            return HttpResponseRedirect("/fights/")
        else:
            form = CreateForm()
            return render(request, 'create_boxer.html', {'form': form})

```

AddBill.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>add Bill</title>
</head>

```

```

<body>
  <form method="post">{% csrf_token %}
    <div class="form-group">
      {% for el in errors %}
        <div>
          <strong>{{ el }}</strong>
        </div>
      {% endfor %}
    </div>
    <p><label>bank_card:
    <input type="number" name="bank_card" value="{{ bank_card }}">
    </label></p>
    <p><label>CVV_CVC_code:
    <input type="number" name="cvv_cvc_code" value="{{ cvv_cvc_code }}">
    </label></p>
    <p><label>sym:
    <input type="number" name="sym" value="{{ sym }}">
    </label></p>
    <button type="submit">Пополнить счет</button>
  </form>
</body>
</html>

```

Base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
  <title>{% block title %}{% endblock %}</title>
</head>
<body>

  <div class="jumbotron" >
    <div class="container" align="center">
      <h1>Бокс</h1>
      <p>{% block header %}{% endblock %}</p>
    </div>
  </div>

  <div>
    {% block body %}{% endblock %}
  </div>
</body>
</html>

```

Create_boxer.html

```

<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
  <!--

```

```

function validate_form ( )
{
    valid = true;
    if ( document.contact_form.idBoxer.value == "" ||
document.contact_form.title.value == "" ||
        document.contact_form.boxer_first_name.value == "" ||
document.contact_form.boxer_last_name.value == "" )
    {
        alert ( "Заполните все поля" );
        valid = false;
    }
    return valid;
}

//-->
</script>
</head>
<body>
    <form name="contact_form" method="POST" onsubmit="return validate_form(
);" enctype="multipart/form-data" >
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" name="send" value="Создать">
    </form>

```

Fight.html

```

{% extends 'base.html' %}

{% block title %}Подробности боя{% endblock %}

{% block header %}
    {% for f in fight %}
        <label>{{ f.boxer1 }} VS {{ f.boxer2 }}</label>
    {% endfor %}
{% endblock %}

{% block body %}
    {% if user.is_authenticated %}
        <p align="right"><span class='a1' >Вы авторизованы.</span><br>
        <span class='a1' >Здравствуй, {{user.username}}</span><br>
        {% for b in bills %}
            <span class='a1' >Ваш счет: {{b.bill}}</span><br>
            <span class='a1'><a href="{% url 'add_bill' %}"
title="Пополнить счет"> Пополнить счет </a></span><br>
        {% endfor %}
        <span class='a1'><a href="{% url 'logout' %}" title="Выход">
Выход </a></span><br></p>
    {% else %}
        <p align="right"><span class='a1' >Вы не авторизованы</span><br>
        <span class='a1'><a href="{% url 'login' %}" title="Вход"> Вход
</a></span><br>
        <span class='a1'><a href="{% url 'registration' %}"
title="Регистрация"> Регистрация </a></span></p>
    {% endif %}
    <a href="{% url 'fights' %}" type="button" class="btn btn-success"><-
Назад к списку</a><br><br>
    {% for f in fight %}
        <label>Дата поединка: {{f.date}}</label><br>
        <label>Время: {{f.time}}</label><br>
        <hr/>
        {% for b in boxer %}
            {% if f.idBoxer1 == b.idBoxer %}
                <label>1-й боксер</label><br>
                <label>Весовая категория: {{ b.title }}</label><br>

```

```

        <label>Фамилия: {{ b.boxer_last_name }}</label><br>
        <label>Имя: {{ b.boxer_first_name }}</label><br>
        <hr/>
    {% endif %}
    {% if f.idBoxer2 == b.idBoxer %}
        <label>2-й боксер</label><br>
        <label>Весовая категория: {{ b.title }}</label><br>
        <label>Фамилия: {{ b.boxer_last_name }}</label><br>
        <label>Имя: {{ b.boxer_first_name }}</label><br>
        <hr/>
        <center><a href="{% url 'make_bet' f.idFight %}"
type="button" class="btn btn-success"> Сделать ставку </a></center>
    {% endif %}
    {% endfor %}
    {% endfor %}
    <label>Уже сделали ставку:</label><br>
    {% for l in logs1 %}
        <label>{{ l.userLogin2 }}</label><br>
    {% endfor %}
{% endblock %}

```

Index.html

```

{% extends 'base.html' %}

{% block header %}Расписание боев
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></scri
pt>
    <style>
        .block{
            width: 200px;
            height: 100px;
            background: #ccc;
            margin-bottom: 5px;
        }
        #popup {
            display: none;
            background: #fff;
            position: absolute;
            z-index: 999;
            width: 300px;
            height: 370px;
            top: 50%;
            left: 50%;
            margin-left: -150px;
            margin-top: -150px;
        }
        #hover {
            display: none;
            background: rgba(182, 201, 187, 0.38);
            position: absolute;
            z-index: 998;
            width: 100%;
            height: 100%;
            top: 0;
            left: 0;
        }
    </style>
{% endblock %}

{% block title %}Ближайшие бои{% endblock %}

```

```

<html>
<body>
{% block body %}
    {% if user.is_authenticated %}
        <p align="right"><span class='a1' >Вы авторизованы.</span><br>
        <span class='a1' >Здравствуй, {{user.username}}</span><br>
        {% for b in bills %}
            <span class='a1' >Ваш счет: {{b.bill}}</span><br>
            <span class='a1'><a href="{% url 'add_bill' %}"
title="Пополнить счет"> Пополнить счет </a></span><br>
        {% endfor %}
        <span class='a1'><a href="{% url 'logout' %}" title="Выход">
Выход </a></span><br></p>
        {% if user.is_staff %}
            <center><a href="{% url 'NewFight' %}" type="button"
class="btn btn-success">Новый поединок</a></center><br>
            <center><a href="{% url 'create_boxer' %}" type="button"
class="btn btn-success">Новый боксер</a></center><br>
            <center><button id="button" class="btn btn-success">Модальное
диалоговое окно</button></center>
            <script type="text/javascript">
                $(document).ready(function() {
                    $("#button").click(function () {
                        $("#popup").css("display", "block");
                        $("#hover").css("display", "block");
                    });

                    $("#hover").click(function () {
                        $("#popup").css("display", "none");
                        $("#hover").css("display", "none");
                    });
                });
            </script>
        {% endif %}
    {% else %}
        <p align="right"><span class='a1' >Вы не авторизованы</span><br>
        <span class='a1'><a href="{% url 'login' %}" title="Вход"> Вход
</a></span><br>
        <span class='a1'><a href="{% url 'registration' %}"
title="Регистрация"> Регистрация </a></span></p>
    {% endif %}

    {% for fight in fights %}
        <div id="course">
            <a href="{% url 'fight_url' fight.idFight %}">{{ fight.boxer1 }}
VS {{ fight.boxer2 }}</a><br>
            
            <hr/>
        </div>
    {% endfor %}
    <script type="text/javascript">
        var page = 1;
        $(document).ready(function() {
            $(window).scroll(function() {
                var windowScroll = document.body.scrollTop; //сколько
прокручено

                //var windowHeight = document.body.clientHeight;
                //размер окна

                var documentHeight = $(document).height(); //размер
загруженной части

                console.log(windowScroll + " " + documentHeight);
                if (Math.round((windowScroll)*2) > documentHeight) {

```

```

        //alert("222222");
        $("#wrapper").append('<div
class="block">Картинка</div>')
        /*$.ajax({
            url: '/fights/${page}/',
            type: 'GET',
            success: function (data) {
                course.innerHTML += data;
                //$('#loading_block').hide();
            }
        }) */
    });
});
</script>

<div class="pagination">
    <span class="step-links">
        {% if fights.has_previous %}
            <center><a href="?page={{ fights.previous_page_number
}}">previous</a></center>
        {% endif %}

        {% if fights.has_next %}
            <center><a href="?page={{ fights.next_page_number
}}">next</a></center>
        {% endif %}
    </span>
</div>
<div id="wrapper">
    <div class="block">Картинка</div>
    <div class="block">Картинка</div>
    <div class="block">Картинка</div>
    <div class="block">Картинка</div>
</div>

<div id="popup">
    <form method="post">{% csrf_token %}
        <div class="form-group">
            {% for el in errors %}
                <div>
                    <strong>{{ el }}</strong>
                </div>
            {% endfor %}
        </div>
        <p><label>idFight:
        <input type="number" name="idFight" value="{{ idFight }}">
        </label></p>
        <p><label>date:
        <input type="date" name="date" value="{{ date }}">
        </label></p>
        <p><label>time:
        <input type="time" name="time" value="{{ time }}">
        </label></p>
        <p><label>boxer1:
        <input type="text" name="boxer1" value="{{ boxer1 }}">
        </label></p>
        <p><label>boxer2:
        <input type="text" name="boxer2" value="{{ boxer2 }}">
        </label></p>
        <p><label>idBoxer1:
        <input type="number" name="idBoxer1" value="{{ idBoxer1 }}">
        </label></p>
        <p><label>idBoxer2:

```



```

        <input type="number" name="idBoxer2" value="{{ idBoxer2 }}">
      </label></p>
      <button type="submit">Создать</button>
    </form>
  </div>
</div id="hover"></div>
{% endblock %}
</body>
</html>

```

Login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
</head>
<body>
  <form method="post">{% csrf_token %}
    <div class="form-group">
      {% for el in errors %}
        <div>
          <strong>{{ el }}</strong>
        </div>
      {% endfor %}
    </div>
    <p><label>Логин:</label>
    <input type="text" name="username" value="{{ username }}">
    </label></p>
    <p><label>Пароль:</label>
    <input type="text" name="password">
    </label></p>
    <button type="submit">Войти</button>
  </form>
</body>
</html>

```

makeBet.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>add Bill</title>
</head>
<body>
  <form method="post">{% csrf_token %}
    <div class="form-group">
      {% for el in errors %}
        <div>
          <strong>{{ el }}</strong>
        </div>
      {% endfor %}
    </div>
    <p><label>Размер ставки:</label>
    <input type="number" name="size_bet" value="{{ size_bet }}">
    </label></p>
    <p><label>Кто победит?</label><br>
    <input name="boxer" type="radio" value="boxer1">{{ box1.boxer1 }}<br>
    <input name="boxer" type="radio" value="boxer2">{{ box1.boxer2 }}<br>
  </form>

```

```

        <button type="submit"> Сделать ставку </button>
    </form>
</body>
</html>

```

newFight.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>NewFight</title>
</head>
<body>
    <form method="post">{% csrf_token %}
        <div class="form-group">
            {% for el in errors %}
            <div>
                <strong>{{ el }}</strong>
            </div>
            {% endfor %}
        </div>
        <p><label>idFight:
        <input type="number" name="idFight" value="{{ idFight }}">
        </label></p>
        <p><label>date:
        <input type="date" name="date" value="{{ date }}">
        </label></p>
        <p><label>time:
        <input type="time" name="time" value="{{ time }}">
        </label></p>
        <p><label>boxer1:
        <input type="text" name="boxer1" value="{{ boxer1 }}">
        </label></p>
        <p><label>boxer2:
        <input type="text" name="boxer2" value="{{ boxer2 }}">
        </label></p>
        <p><label>idBoxer1:
        <input type="number" name="idBoxer1" value="{{ idBoxer1 }}">
        </label></p>
        <p><label>idBoxer2:
        <input type="number" name="idBoxer2" value="{{ idBoxer2 }}">
        </label></p>
        <button type="submit">Создать</button>
    </form>
</body>
</html>

```

Registration.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Registration</title>
</head>
<body>
    <form method="post">{% csrf_token %}
        <div class="form-group">
            {% for el in errors %}
            <div>

```

```

        <strong>{{ el }}</strong>
    </div>
    {% endfor %}
</div>
<p><label>Логин:
<input type="text" name="username" value="{{ username }}">
</label></p>
<p><label>Пароль:
<input type="text" name="password">
</label></p>
<p><label>Повторите ввод:
<input type="text" name="password2">
</label></p>
<p><label>E-mail:
<input type="text" name="E_mail" value="{{ E_mail }}">
</label></p>
<p><label>Фамилия:
<input type="text" name="surname" value="{{ surname }}">
</label></p>
<p><label>Имя:
<input type="text" name="name" value="{{ name }}">
</label></p>
<button type="submit">Зарегистрировать</button>
</form>
</body>
</html>

```

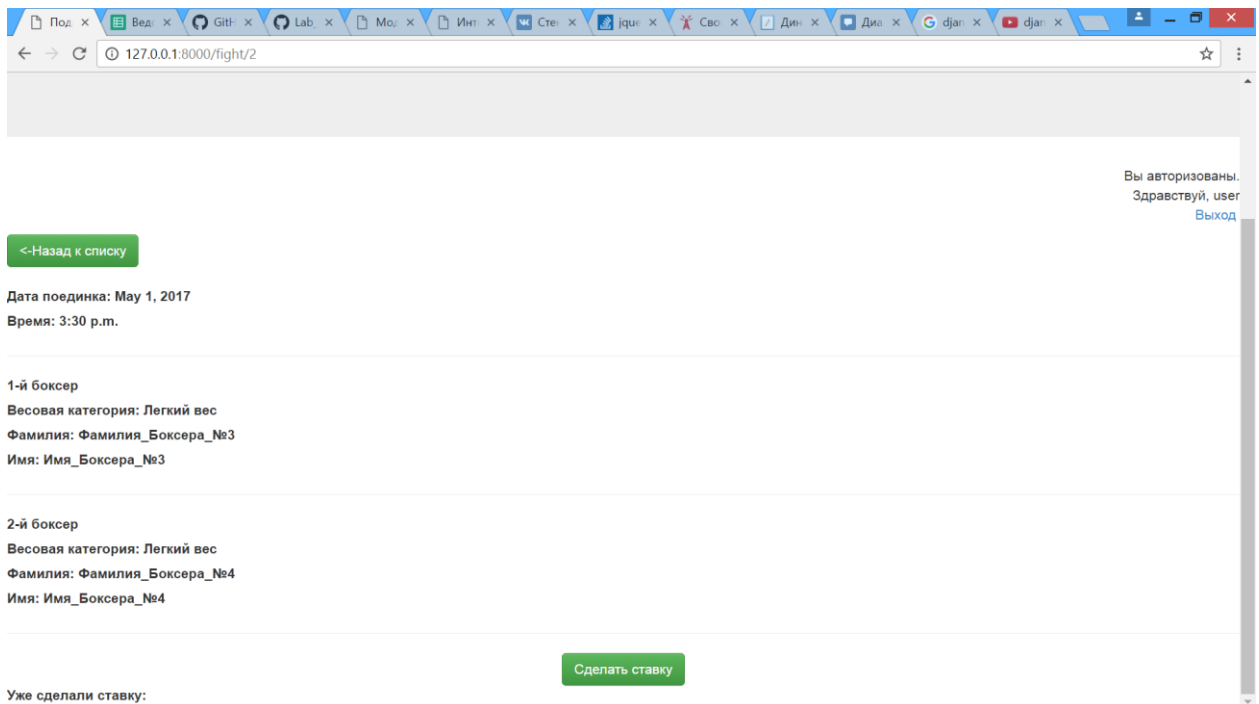
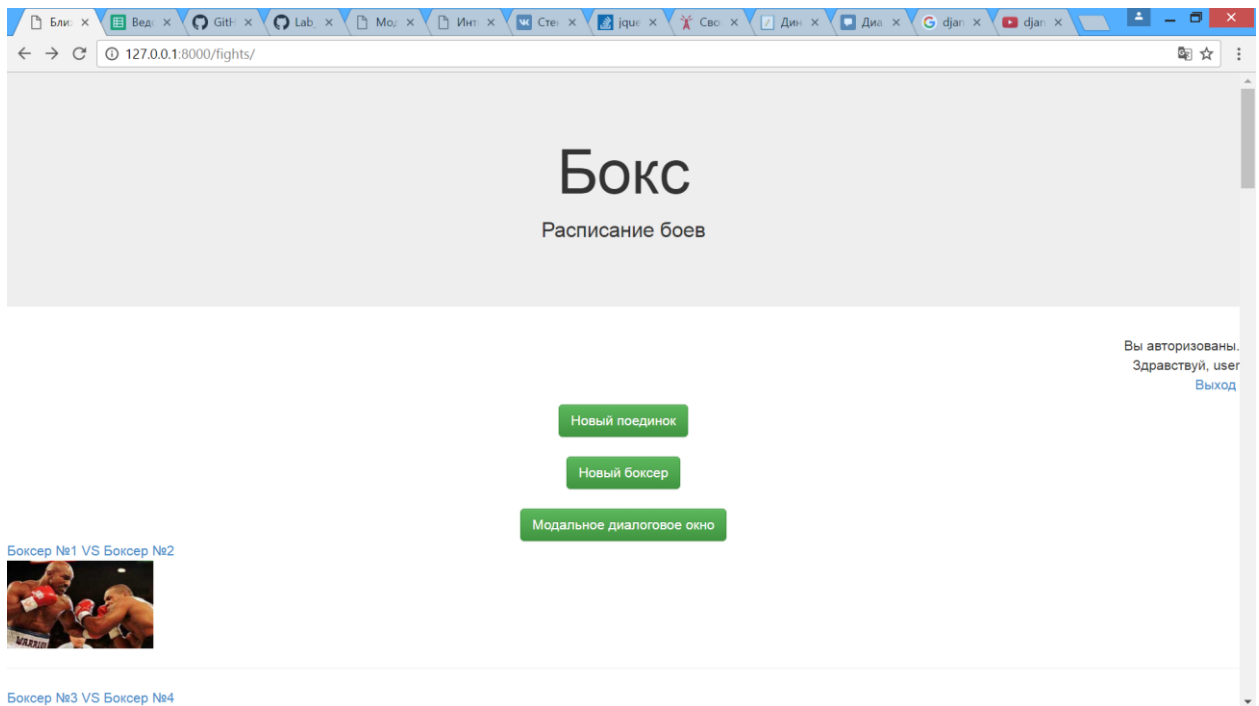
Результаты работы программы:

The screenshot shows the Django administration interface in a web browser. The address bar indicates the URL is 127.0.0.1:8000/admin/. The page title is "Django administration" and it includes a welcome message for the user, along with links to view the site, change password, and log out.

The main content area is titled "Site administration" and contains two primary sections:

- Authentication and Authorization:** A table with two rows: "Groups" and "Users". Each row has a green plus icon and the text "Add" followed by a yellow pencil icon and the text "Change".
- Users:** A table with five rows: "Bets", "Bills", "Boxers", "Fights", and "Users". Each row has a green plus icon and the text "Add" followed by a yellow pencil icon and the text "Change".

On the right side of the page, there is a "Recent actions" section. It contains a "My actions" subsection which currently displays "None available".



add x Bed x Git x Lab x Mo x Ин x Cre x jqu x Сво x Дин x Дил x djan x djan x

127.0.0.1:8000/makeBet/2

Размер ставки:

Кто победит?

☐ Боксер №3

☐ Боксер №4

Regi x Bed x Git x Lab x Mo x Ин x Cre x jqu x Сво x Дин x Дил x djan x djan x

127.0.0.1:8000/registration/

Логин:

Пароль:

Повторите ввод:

Е-mail:

Фамилия:

Имя:

Logi x

Вед x

Git x

Lab x

Мо x

Ин x

Ст x

ique x

Све x

Дин x

Диа x

djan x

djan x

←

→

↻

127.0.0.1:8000/login/

🔍

☆

⋮

Логин:

Пароль:

Войти
