

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Домашнее задание по курсу:

«Разработка Интернет Приложений»

Исполнитель:

Студент группы РТ5-51

Чечнев А.А.

Преподаватель:

Гапанюк Ю. Е.

« ____ » _____



Москва 2017 г.

Задание и порядок выполнения

Разработать вебсервис на базе технологий: Python, Django, JS, MySQL

№	Субъект	Отношение	Объект
21	Пользователь	Ставка	Лошадь

Исходный код:

settings.py

```
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/1.11/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '&y0j%n0jmgb#@o@urg31^arzf=-5g@eep* (^b+++t8v@$y$#4) '

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

ALLOWED_HOSTS = [
    '127.0.0.1', '192.168.1.101', '192.168.1.102', '192.168.1.103', '192.168.1.104',
    '192.168.1.107', '37.204.115.123'
]

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'mainapp.apps.MainappConfig',
    'authorisation.apps.AuthorisationConfig',
]

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

ROOT_URLCONF = 'dz.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 = 'dz.wsgi.application'

# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'sweepstake1',
        'USER': 'root',
        'PASSWORD': ' ',
        'PORT': 3306,
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.11/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/1.11/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/1.11/howto/static-files/

STATIC_URL = '/static/'

MEDIA_ROOT = os.path.join(BASE_DIR, 'mainapp/media')
MEDIA_URL = '/media/'

```

urls.py

```

from django.conf.urls import url, include
from django.contrib import admin
from django.core.urlresolvers import reverse

```

```

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', include('mainapp.urls')),
    url(r'^', include('authorisation.urls')),
]

```

authorisation
urls.py

```

from django.conf.urls import url, include
from django.contrib import admin
from .views import *
from django.views.generic.base import RedirectView

```

```

urlpatterns = [

    url(r'^login/$', LoginFormView.as_view(), name="login_url"),
    url(r'^registration/$', registration, name="registration_url"),
    url(r'^logout/$', LogoutView.as_view(), name="logout_url"),
    url(r'^$', RedirectView.as_view(url='/login/', permanent=False),
name='index'),

]

```

views.py

```

from django.shortcuts import render
from django.contrib.auth import login, logout
from django.views.generic.edit import View, FormView

```

```

from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth.models import User
from django.core.urlresolvers import reverse
from django.http import HttpResponseRedirect
from .forms import *

class LoginFormView(FormView):
    def get(self, request, *args, **kwargs):
        if request.user.is_authenticated:
            return HttpResponseRedirect(reverse("runs_url"))
        else:
            return super(LoginFormView, self).get(self, request)

    form_class = AuthenticationForm
    template_name = "login.html"

    success_url = "/login"

    def form_valid(self, form):
        self.user = form.get_user()
        login(self.request, self.user)

        return super(LoginFormView, self).form_valid(form)

    def registration(request):
        form = RegistrationForm(request.POST or None)
        # password_error = ''
        if request.method == 'POST':
            if form.is_valid():
                user =
                User.objects.create_user(username=form.cleaned_data['username'],
                email=form.cleaned_data['email'],
                password=form.cleaned_data['password'],
                first_name=form.cleaned_data['firstname'],
                last_name=form.cleaned_data['surname'])
                # ...
                return HttpResponseRedirect(reverse("login_url"))

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

class LogoutView(View):
    def get(self, request):
        # Выполняем выход для пользователя, запросившего данное
        # представление.
        logout(request)

        # После чего, перенаправляем пользователя на главную страницу.
        return HttpResponseRedirect(reverse("login_url"))

```

```

forms.py
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm

class RegistrationForm(forms.Form):
    username = forms.CharField(min_length=5, label='Логин',
widget=forms.TextInput)
    password = forms.CharField(min_length=6, widget=forms.PasswordInput,
label='Пароль')
    password2 = forms.CharField(min_length=6, widget=forms.PasswordInput,
label='Повторите пароль')
    email = forms.EmailField(widget=forms.EmailInput, label='E-mail')
    firstname = forms.CharField(label='Имя')
    surname = forms.CharField(label='Фамилия')

    def is_valid(self):
        b = True
        if not super().is_valid():
            b = False

        if self.cleaned_data['password'] != self.cleaned_data['password2']:
            self.add_error('password', 'Пароли не совпадают')
            b = False

        if
User.objects.filter(username=self.cleaned_data['username']).exists():
            self.add_error('username', 'Такой пользователь уже существует')
            b = False
        if self.errors.get('email'):
            self.errors['email'] = ['Введите корректный email']
            b = False
        return b

```

mainapp

```

urls.py
from django.conf.urls import url, include
from django.contrib import admin
from .views import *
from django.contrib.auth.decorators import login_required
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [

    url(r'^runs/add$', login_required(run_add_view), name="run_add_url"),
    url(r'^runs$', login_required(RunView.as_view()), name="runs_url"),
    url(r'^runs1$', login_required(RunView1.as_view()), name="runs1_url"),

    url(r'^horses/$', login_required(horse), name="horse_url"),

    # url(r'^run', login_required(run view), name="run url"),
    # url(r'^base', login_required(view1)),
]

if settings.DEBUG:

```

```
urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

```
views.py
from django.shortcuts import render
from django.core import serializers
from django.views.generic import ListView, View, FormView
from django.contrib.auth.models import User
from django.core.urlresolvers import reverse
from django.http import HttpResponseRedirect, HttpResponse
from .models import *
from .forms import *

class RunView(ListView):
    # form_class = RunForm
    form = RunForm
    model = Run
    template_name = "runs_list.html"
    context_object_name = "runs"
    # queryset = ['id', 'date', 'time', 'place']
    paginate_by = 10

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['form'] = self.form
        return context

    def get(self, self, request, *args, **kwargs):
        run_id = request.GET.get('id')
        if not run_id:
            return super(RunView, self).get(request, *args, **kwargs)

        else:
            runs = Run.objects.filter(id=run_id)
            self.horse_id = request.GET.get('horse')
            horse_id = self.horse_id
            money_error = kwargs.get('money_error')
            return render(request, "run.html", locals())

    def post(self, self, request):
        run_id = request.GET.get('id')
        if not run_id:
            form = self.form(request.POST, request.FILES)
            if form.is_valid():
                instance = form.save(commit=False)
                instance.save()
                return HttpResponseRedirect(reverse('runs_url'))
            return super(RunView, self).get(request)
        else:
            runs = Run.objects.filter(id=run_id)
            u = User.objects.get(id=request.user.id)
            h = HorseInRun.objects.get(horse=request.GET.get('horse'),
run=run_id)

            money = request.POST.get('money')
            money_error = ""
            if not money or money == "0":
                money_error = 'Введите сумму ставки'
            else:
```

```
        if float(money) > float(u.cash):
            money_error = 'У вас на счету всего ' + str(u.cash) + '
рублей'
```

```
        else:
            b = Bet(user=u, amount=money, horse_in_run=h)
            u.cash = u.cash - float(money)
            u.save()
            b.save()
        return self.get(request, money_error=money_error)
        # return render(request, "run.html", locals())
```

```
class RunView1(ListView):
    form_class = RunForm
    model = Run
    template_name = "runs_list1.html"
    context_object_name = "runs"
    paginate_by = 1
```

```
def run_add_view(request):
    form = RunForm(request.POST or None, request.FILES or None)
    print(form.as_p())
    if request.method == 'POST':
        if form.is_valid():
            instance = form.save(commit=False)
            instance.save()

            return HttpResponseRedirect(reverse('runs_url'))
    return render(request, "run_add.html", locals())
```

```
def view1(requests):
    title = "main"
    return render(requests, "base.html", locals())
```

```
class HorseListView(ListView):
    model = Horse
    template_name = "horses.html"
    context_object_name = "obj_list"

    def get(self, request, *args, **kwargs):
        self.id = request.GET.get('id')
        return super(HorseListView, self).get(request, *args, **kwargs)

    def get_queryset(self):
        if self.id:
            return Horse.objects.filter(id=self.id)
        else:
            return Horse.objects.all()
```

```
class HorseListView1(ListView):
    model = Horse
    template_name = "horse.html"
    context_object_name = "obj_list"

    def get(self, request, *args, **kwargs):
        self.id = request.GET.get('id')
```



```

        return super(HorseListView1, self).get(request, *args, **kwargs)

    def get_queryset(self):
        if self.id:
            return Horse.objects.filter(id=self.id)
        else:
            return Horse.objects.all()

    def horse(request):
        id = request.GET.get('id')

        if id:
            return render(request, "horse.html", {'horse':
Horse.objects.get(id=id)})
        else:
            return render(request, "horses.html", {'horses':
Horse.objects.all()})

```

```

models.py
from django.db import models
from django.db.models import Q
from django.contrib.auth.models import User
from datetime import *

class Jockey(models.Model):
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    birth_date = models.DateField(verbose_name='Дата рождения')

    def age(self):
        today = datetime.utcnow()
        # born = datetime.strptime(self.birth_date, "%d.%m.%Y")
        born = self.birth_date
        return today.year - born.year - ((today.month, today.day) <
(born.month, born.day))

    def __str__(self):
        return '%s %s' % (str(self.last_name), str(self.first_name))

class Horse(models.Model):
    name = models.CharField(max_length=100, unique=True, null=False,
verbose_name='Имя')
    age = models.IntegerField(null=False, verbose_name='Возраст')
    photo = models.ImageField(null=True, blank=True, default=None,
verbose_name='Фото')
    jockey = models.ForeignKey(Jockey, verbose_name='Жockey', default=None)

    def __str__(self):
        return str(self.id) + " _ " + self.name

    def get_jockey(self):
        return str(self.jockey.last_name) + str(self.jockey.first_name)

    get_jockey.short_description = 'Жockey'

class Run(models.Model):

```

```

date = models.DateField(verbose_name='Дата')
time = models.TimeField(verbose_name='Время')
place = models.CharField(max_length=255, verbose_name='Место')
horse = models.ManyToManyField(Horse, through='HorseInRun',
through_fields=('run', 'horse'))
place_image = models.ImageField(null=True, blank=True, default=None,
verbose_name='место')

def __str__(self):
    return '%s ____ %s ____ %s' % (str(self.id), str(self.date),
str(self.time))

def get_horses(self):
    return [{ 'id': horse.id, 'name': horse.name, 'age': horse.age} for
horse in self.horse.all()]

get_horses.short_description = 'Участники'

class HorseInRun(models.Model):
    horse = models.ForeignKey(Horse, on_delete=models.CASCADE,
verbose_name='Лошадь')
    run = models.ForeignKey(Run, on_delete=models.CASCADE,
verbose_name='Забор')
    user = models.ManyToManyField(User, through='Bet',
through_fields=('horse_in_run', 'user'))

def __str__(self):
    return '%s %s' % (str(self.run.date), str(self.horse.name))

class Bet(models.Model):
    horse_in_run = models.ForeignKey(HorseInRun, verbose_name='Ставка на
лошадь')
    user = models.ForeignKey(User, verbose_name='Пользователь')
    amount = models.DecimalField(max_digits=15, decimal_places=2,
verbose_name='Ставка')

def __str__(self):
    return '%s %s: %s' % (str(self.user.username),
str(self.horse_in_run.run.id), str(self.horse_in_run.horse.name))

```

forms.py

```

from django import forms
from django.forms import ModelForm
from django.contrib.auth.models import User
from .models import *

```

```

class RunForm(ModelForm):

```

```

    class Meta:
        model = Run
        exclude = ['horse']

```

admin.py

```

from django.contrib import admin
from .models import *
from django.contrib.auth.admin import UserAdmin

```

```

class HorseAdmin(admin.ModelAdmin):
    # list_display = ["id", "name"] # Выводит 2 поля

    list_display = [field.name for field in Horse._meta.fields] # Выводит
все поля

    # list_filter = ["good_id"]
# Фильтр по полям

    class Meta:
        model = Horse

class RunAdmin(admin.ModelAdmin):
    list_display = [field.name for field in Run._meta.fields] # Выводит
все поля

    fieldsets = [
        ('Дата забера', {'fields': ['date', 'time']}),
        ('Место забера', {'fields': ['place', 'place_image']}),
    ]

    # list_filter = ["good_id"]
# Фильтр по полям

    class Meta:
        model = Run

class HorseInRunAdmin(admin.ModelAdmin):
    list_display = ["id", "run", "horse"] # Выводит 2 поля

    # list_display = [field.name for field in HorseInRun._meta.fields] #
Выводит все поля

    list_filter = ["run"] # Фильтр по полям

    class Meta:
        model = HorseInRun

class BetAdmin(admin.ModelAdmin):
    list_display = ["user", "horse_in_run", "amount"] # Выводит 2 поля

    list_filter = ["user", "horse_in_run"] # Фильтр по полям

    class Meta:
        model = Bet

admin.site.register(Horse, HorseAdmin)
admin.site.register(Run, RunAdmin)
admin.site.register(HorseInRun, HorseInRunAdmin)
admin.site.register(Bet, BetAdmin)
admin.site.register(Jockey)

```

main.css

```
@import url('https://fonts.googleapis.com/css?family=Lato|Raleway');
@import "bootstrap.css";

body{
    background-color: #f2f2f2;
    font-family: "Lato", serif;
    font-weight: 300;
    font-size: 16px;
    color: #555;

    -webkit-font-smoothing: antialiased;
    -webkit-overflow-scrolling: touch;

    margin-top: 70px;
}

h1, h2, h5, h6{
    font-family: "Raleway", serif;
    font-weight: 300;
    color: #e70216;
}

h3{
    font-family: "Arial", serif;
    font-weight: bolder;
    color: rgba(33, 166, 3, 0.58);

    text-decoration: #0cfaff;
}

h4{
    font-family: "Ebrima", serif;
    font-weight: bolder;
    color: rgba(59, 21, 255, 0.92);

    text-decoration: #3b15ff;
}

p{
    line-height: 28px;
    margin-bottom: 25px;
}

.centered{
    text-align: center;
}

a{
    color: #f85c37;
}

a:hover, a:focus{
    color: #7b7b7b;
    text-decoration: none;
}

hr{
    display: block;
    height: 1px;
```

```

border: 0;
border-top: 1px solid #ccc;
margin: 1em 0;
padding: 0;
}

.navbar{
font-weight: 800;
font-size: 14px;
padding-top: 15px;
padding-bottom: 15px;
}
.navbar-inverse{
background: rgba(0, 0, 0, 0.58);
border-color: #fbff00;
border-bottom-style: ridge;
}

.navbar-inverse .navbar.nav > .active > a{
background-color: #ff7878;
}

.navbar-inverse .navbar-brand{
color: #999;
font-weight: bolder;
font-size: 22px;
letter-spacing: 1px;
}

{% load static %}

#headerwrap1{
/*background: url(lg.jpg) no-repeat center top;*/
background: #e5fffd;
margin: 40px auto 20px;
padding-right: 50px;
padding-left: 30px;
text-align: right;

background-position: center center;

min-height: 10px;
width: 100%;
}

#headerwrap{
/*background: url(lg.jpg) no-repeat center top;*/
background: #e5fffd;
margin-top: 40px;
margin-bottom: 20px;
text-align: right;
text-decoration: dotted;
background-origin: border-box;
backface-visibility: hidden;
background-position: center center;
min-height: 700px;
width: 100%;

background-size: 100%;

```

```

        -webkit-background-size: cover;
        -moz-background-size: cover;
    }

    #header1 {
        background: rgba(124, 255, 230, 0.53);
        text-align: center;
        padding-top: 15px;
        min-height: 100px;
    }

    header1 h4{
        color: deeppink;
    }

    table {
        border-collapse: collapse;
        line-height: 3;
        font-family: "Lato", "Mistral", sans-serif;
        background: radial-gradient(farthest-corner at 50% 50%, #f3fff6,
rgba(219, 248, 209, 0.7));
        color: #0C213B;
    }

    th {
        text-align: center;
        padding: 20px;
        border: 1px solid #61e2e2;
    }

    td {
        text-align: center;
        font-size: 0.8em;
        padding: 10px 12px;
        border: 1px solid #A9E2CC;
    }

    table td a,h4{
        color: darkblue;
        font-family: "Lato", "Mistral", sans-serif;
    }

    table tr{
        min-width: 200px;
    }

    table tr a{
        display: block;
        width: auto;
        height: 100%;
        white-space: pre-line;
        min-width: 200px;
    }

    caption{
        padding: 10px;
    }

```

```

        font-size: 30px;
    }

    .container1 {
        margin: 80px auto;
        width: 90%;
        max-width: 800px;
    }

    .registration {
        background-color: #fff;
        text-align: right;
        padding-right: 150px;
        max-width: 600px;
        margin: 0 auto;
        margin-top: 1px;
        padding-bottom: 20px;
        box-shadow: 1px 1px 5px 0 rgba(0,0,0,0.3);
        border-bottom: 5px solid #ffdb3a;
    }

    .dropdown-toggle {
        margin: 7px;
        background-color: inherit;
        border: none;
    }

    .dropdown-content {
        width: 150px;
        background-color: #444444;
        border: none;
        border-radius: 10px;
        top: 100%;
        display: none;
        position: absolute;
        margin-left: -100px;
        margin-top: -20px;
        font-size: 15px;
        box-shadow: 10px 10px 10px rgba(0,0,0,0.5);
    }

    .dropdown-content a {
        color: grey;
        display: inline-block;
        padding: 5px;
        margin: 10px;
    }

    .dropdown-content a:hover {
        background-color: inherit;
        color: white;
        font-size: 17px;
        padding: 3px;
        padding-bottom: 4px;
    }

    .dropdown:hover .dropdown-content {
        display: inline-block;
        border-color: #999999;
        background-color: black;
    }

```

```

.dropdown-content p {
    margin-left: 25px;
    margin-right: 25px;
    display: inline-block;
    text-align: center;
    color: #f2f2f2;
}

.dropdown-content span {
    padding: 2px;
    margin-left: 15px;
    margin-right: 25px;
    display: inline-block;
    text-align: center;
    color: white;
}

```

templates

base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    {% block scripts %}{% endblock %}
    <meta charset="UTF-8">
    {% load static %}

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></sc
ript>
    <script src={% static "Bootstrap/js/bootstrap.min.js" %}></script>
    <script src="{% static 'Bootstrap/js/jquery.waypoints.min.js'
%}"></script>
    <script src="{% static 'Bootstrap/js/infinite.min.js' %}"></script>

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{{ title }}</title>
    <link rel="stylesheet" href={% static 'Bootstrap/css/bootstrap.min.css'
%}>
    <link rel="stylesheet" href={% static 'Bootstrap/css/font-
awesome.min.css'%}>
    <link rel="stylesheet" href={% static 'Bootstrap/css/main.css'%}>
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                    <div class="col-lg-1 col-lg-offset-0">

                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>

                    </div>
                </button>

```



```

        <a class="navbar-brand" href="#"><i class="fa fa-ellipsis-h"></i>{{
head_text }}</a>
    </div>
    <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav navbar-right">

            {% block head_buttons %}{% endblock %}

        <li>
            <div class="dropdown">
                <button class="dropdown-toggle" data-
toggle="dropdown">
                    {#                               <img src="" width="30" height="30"
alt=""/>#}
                    <i class="fa fa-user"></i>
                </button>
                <div class="dropdown-content">
                    {% if request.user.is_authenticated %}
                    <span>Здравствуйте, {{
request.user.username }}</span>
                    <a href="{% url 'logout_url'
%}">Выйти</a>
                    {% else %}
                    <a href="{% url 'login_url'
%}">Вход</a>
                    <a href="{% url 'registration_url'
%}">Регистрация</a>
                    {% endif %}
                </div>
            </div>
        </li>
        {#                               {% endif %}#}
    </ul>
</div>
</div>
</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></sc
ript>
<script src={% static "Bootstrap/js/bootstrap.min.js" %}></script>

{% block body1 %}{% endblock %}

</body>

</html>

```

```

login.html
{% extends "base.html" %}

{% block body1 %}
    <div class="container1">

```

```

        <div class="registration">
        <br><br>
        <form action="" method="post">
            {% csrf_token %}

            {{ form.as_p }}

            <button type="submit">Войти</button>

        </form>
        <br>

        <form method="get" action={% url "registration_url" %}>
            <button>Регистрация</button>
        </form>

    </div>
</div>

{% #</div>#}
{% endblock %}

```

registration.html

```

{% extends "base.html" %}

{% block body1 %}
    <div class="container1">

        <div class="registration">
        <br><br>
        <form action="" method="post">
            {% csrf_token %}

            {{ form.as_p }}

            <button type="submit">Войти</button>

        </form>
        <br>

        <form method="get" action={% url "registration_url" %}>
            <button>Регистрация</button>
        </form>

    </div>
</div>

{% #</div>#}
{% endblock %}

```

run.html

```

{% extends 'base.html' %}

{% block head_buttons %}
{% endblock %}

```

```

{% block body1 %}

<div class="container">
  <div class="row">
    <br><br>
    {% if request.user.is_authenticated %}
      <div class="col-lg-1">
        <a href={% url "runs_url" %}>
          <div class="fa fa-arrow-circle-left"></div>
        </a>
      </div>
      <div class="col-lg-8">
        {% for run in runs %}
          <table>
            <caption class="centered"><h3>Забег номер {{ run.id
}}</h3></caption>
            <tr>
              <th>Дата</th>
              <th>Время</th>
              <th>Место</th>
            </tr>
            <tr>
              <td><h4>{{ run.date }}</h4></td>
              <td><h4>{{ run.time }}</h4></td>
              <td><h4>{{ run.place }}</h4></td>
            </tr>
          </table>
        </div>
        <div class="col-lg-3">
          <div>Участники: </div>
          {% for horse in run.get_horses %}
            <h4>
              <a href={% url "runs_url" %}?id={{
run.id }}&horse={{ horse.id }}>
                {{ horse.id }} - {{
horse.name }} - {{ horse.age }} лет
              </a>
            </h4>
          {% endfor %}
        </div>
        {% endif %}
        {% if horse_id != None %}
          <form method="POST">
            {% csrf_token %}
            <label>
              Сумма ставки <div class="h4"> {{ money_error }}
            </div>
            </label>
            <div class="form-block">
              <input type="text" name="money">
            </div>
            <button type="submit">Сделать ставку</button>
          </form>
        {% endif %}
      </div>
    {% endif %}
  </div>
</div>

```

```

        <br>
    {% endif %}
</div>

</div>

{% endblock %}

run_add.html
{% extends 'base.html' %}

{% block scripts %}
    <script>
        function Validate(obj) {
            var date=obj.date.value.split('-');
            var time=obj.time.value.split(':');
            var errors="";

            if (!(date[0]<10000 && date[1]<24 && date[2]<32)) {
                errors+="Указана неверная дата рождения[гггг-мм-дд]!!\n";
            }

            if (!(time[0]< 24 && time[1]<60)){
                errors+="Неверное время\n"
            }

            if(errors==="")
                return true;

            else {
                alert(errors);
                return false;
            }
        }
    </script>

    <script>
        var form = document.querySelector('.formWithValidation');
        var validateBtn = form.querySelector('.validateBtn');

        form.addEventListener('submit', function (event) {
            event.preventDefault()
        })
    </script>
{% endblock %}

{% block head_buttons %}
    <li><a href={% url "runs_url" %}>К заберам</a></li>
    <li><a href={% url "runs1_url" %}>К заберам1</a></li>
    {#
        <li><a href={% url "good numb_url" 1%}><i class="fa fa-envelope-o"
        aria-hidden="true"></i></a></li>#}
{% endblock %}

```

```
{% block body1 %}

    <div class="container1 right">
        <div class="row">
            {#
                <div class="col-lg-5 col-lg-offset-3">#}
                <div class="registration">
                    <h2>{{ main_text }}</h2>
                    <br><br>

                    <form OnSubmit='return Validate(this);' action="" method="post"
class="formWithValidation">{% csrf_token %}
                        <h3 class="right">
                            {{ form.as_p }}
                        </h3>

                        <div>
                            <button type="submit" class="btn btn-primary
validateBtn">Добавить!!!!</button>
                            <br><br>
                        {#
                            {% if result != '' %}#}
                            {#
                                <h1>#}
                            {#
                                {% csrf_token %}#}
                            {#
                                {{ result }}#}
                            {#
                                </h1>#}
                            {#
                                {% endif %}#}
                        </div>
                    </form>
                </div>
            </div>
        {#
            </div>#}

{% endblock %}
```

runs_list.html

```
{% extends 'base.html' %}

{% block head_buttons %}

    {% if request.user.is_staff %}
        <li><a href={% url "run_add_url" %}>Добавить забег</a></li>
        <li><a href="#course-add-modal" data-toggle="modal">
            <div class="add-button">+</div>
        </a></li>
        <li><a href="">Лошади</a></li>
    {% endif %}

{% endblock %}

{% block body1 %}

<div class="container">
    <div class="row">
        {% if request.user.is_authenticated %}
            <div class="col-lg-12">
                <table>

                    <caption class="fa-5x centered"><a href={% url
"runs1_url" %}> Забег</a></caption>
```

```

        <tr>
            <th>Номер</th>
            <th>Дата</th>
            <th>Время</th>
            <th>Место</th>
            <th>Фото</th>
        </tr>

        {% for run in runs %}
            <tr>
                <td><h4><a href="{% url 'runs_url' %}"?id={{
run.id }}> {{ run.id }} </a> </h4> </td>
                <td><h4><a href="{% url 'runs_url' %}"?id={{
run.id }}> {{ run.date }} </a></h4></td>
                <td><h4><a href="{% url 'runs_url' %}"?id={{
run.id }}> {{ run.time }} </a></h4></td>
                <td><h4><a href="{% url 'runs_url' %}"?id={{
run.id }}> {{ run.place }} </a></h4></td>
                <td>
                    {% if run.place_image %}
                        
                    {% endif %}
                </td>
            </tr>
        {% endfor %}
    </table>
    <span class="page-links">
        {% if page_obj.has_previous %}
            <a class="aquo" href="/runs?page={{
page_obj.previous_page_number }}">&laquo;</a>
        {% else %}
            <span class="unavailable" href="">&laquo;</span>
        {% endif %}

        {% for page in page_obj.paginator.page_range%}
            {% if page %}
                {% ifequal page page_obj.number %}
                    <span class="current-page" href="">{{ page
}}</span>

                {% else %}
                    <a href="/runs?page={{ page }}"
class="page">{{ page }}</a>
                {% endifequal %}
            {% endif %}
        {% endfor %}

        {% if page_obj.has_next %}
            <a class="aquo" href="/runs?page={{
page_obj.next_page_number }}">&raquo;</a>
        {% else %}
            <span class="unavailable" href="">&raquo;</span>
        {% endif %}
        <span hidden id="per-page">{{
page_obj.paginator.per_page }}</span>
    </span>
</div>
{% endif %}
</div>
</div>

```

```

<div id="course-add-modal" class="modal fade">
  <div class="modal-dialog">
    <div class="modal-content">

      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
aria-hidden="true">×</button>
        <h2 class="modal-title">Добавление нового забега</h2>
      </div>

      <div class="modal-body">
        <form method="POST" enctype="multipart/form-data">
          <div class="add-form container">
            <div class="">
              {% csrf_token %}
              {{ form.as_p }}
            </div>
            <button type="submit" class="btn btn-
primary">Добавить!!!!</button>
            <br><br>
          </div>
        </form>
      </div>

      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Закрыть</button>
      </div>
    </div>
  </div>
</div>
{% endblock %}

```

runs_list1.html

```

{% extends 'base.html' %}

{% block scripts %}

  <script src="scroll-startstop.events.jquery.js"
type="text/javascript"></script>

{% endblock %}

{% block head_buttons %}
  {% if request.user.is_staff %}
    <li><a href={% url "run_add_url" %}>Добавить забег</a></li>
    <li><a href="">Лошади</a></li>
  {% endif %}
{% endblock %}

{% block body1 %}

  <div class="container">
    <div class="row">
      <div class="col-lg-12 centered">

```

```

<h2><a href={% url "runs_url" %}> Забег </a></h2>

{% for run in runs %}

<div class="infinite-container">
  <div class="infinite-item">

    <a href={% url "runs_url"%}?id={{ run.id }} >
      <button>
        {% if run.place_image.url != None %}
          
        {% endif %}
        <h4><small> Забег номер: {{ run.id }}
</small></h4>

        <h4><small> {{ run.date }} </small></h4>
        <h4><small> {{ run.time }} </small></h4>
        <h4><small> {{ run.place }} </small></h4>
      </button>
    </a>
    <br><br><br>
  </div>
</div>

{% endfor %}
</div>

<div class="loading" style="display: none;">
  Loading...
</div>

{% if page_obj.has_next %}
  <a class="infinite-more-link" href="?page={{
page_obj.next_page_number }}">More</a>
{% endif %}
</div>
</div>

<div style="display:none;" class="nav_up" id="nav_up"></div>

<script>
  var infinite = new Waypoint.Infinite({
    element: $('.infinite-container')[0],
    onBeforePageLoad: function () {
      $('.loading').show();
    },

    onAfterPageLoad: function ($items) {
      $('.loading').hide();
    }
  });
</script>

{% endblock %}

```