

Московский Государственный Технический Университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

Отчет по лабораторной работе № 7

Курс «Разработка Интернет-приложений»

Выполнил:

студент группы ИУ5-54

(подпись)

Харлашкин А. И.

"__" _____ 2016 г.

Проверил:

Преподаватель каф. ИУ5

(подпись)

Гапанюк Ю. Е.

"__" _____ 2016 г.

Москва, МГТУ – 2016 г.

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.

Исходный код Urls.py

```
from django.conf.urls import url from . import views from bookmaking.views import Registration, Login
```

```
urlpatterns = [ url(r'^$', views.main_page), #url(r'^user/$', user_view.as_view()), url(r'^user/$',
views.user_view), url(r'^registration$', Registration.as_view()), #url(r'^login$', Login.as_view()),
url(r'^login$', views.log_in), url(r'^logout$', views.log_out), ]
```

Views.py

```
from django.shortcuts import render, redirect from django.contrib.auth import authenticate, login,
logout from django.contrib.auth.models import User from .models import User1, RegModel,
LoginModel from django.views import View from pip._vendor import requests from
django.contrib.auth.decorators import login_required from django.utils.safestring import mark_safe
```

```
#class user_view(View): @login_required(login_url='/login') def user_view(request): #def get(request):
users = User1.objects.all() return render (request, 'bookmaking/users.html', {'users': users}) def
log_out(request): logout(request) return redirect("/")
```

```
@login_required(login_url='/login') def main_page(request): return render (request,
'bookmaking/main_page.html', {})
```

```
class Registration(View): def get(self, request): form = RegModel() return render(request,
'bookmaking/registration2.html',{'errors': '', 'form': form.as_p()}) def post (self, request): form =
RegModel(request.POST) if not form.is_valid(): return render(request,
'bookmaking/registration2.html', {'errors': '', 'form': form.as_p()}) us =
User(username=form.cleaned_data['username1'], email=form.cleaned_data['email'],
last_name=form.cleaned_data['surname'], first_name=form.cleaned_data['name'])
us.set_password(form.cleaned_data['password']) us.save(form.cleaned_data['username1'])
user = authenticate(username=form.cleaned_data['username1'],
password=form.cleaned_data['password']) login(request, user) return redirect('/') # class
Registration(View): # def get(self, request): # return render(request,
'bookmaking/registration.html',{'errors': '', 'username': '', 'email': '', 'surname': '', 'name': ''}) # # def
post (self, request): # errors = [] # if request.method == 'POST': # username1 =
request.POST['username'] # password = request.POST['password'] # password2 =
request.POST['password2'] # email = request.POST['email'] # surname =
request.POST['surname'] # name = request.POST['name'] # if not username1: #
errors.append('Введите логин') # if not surname: # errors.append('Введите фамилию')
# if len(username1)<5: # errors.append('Слишком короткий логин') # # if
len(password)<8: # errors.append('Слишком короткий пароль') # if
```

```

password!=password2: #          errors.append('Пароли должны совпадать') #          #          if
len(errors)==0:

#          users=User.objects.filter(username=username1) #          if len(users)>0: #
errors.append('Пользователь с таким именем уже существует') #          else: #
us = User(username=username1, email=email, last_name=surname, first_name=name) #
us.set_password(password) #          us.save() #          if len(errors)>0: #          return
render(request, 'bookmaking/registration.html', {'errors':mark_safe('<br>'.join(errors)),
'username1':username1, 'email':email, 'surname':surname, 'name':name}) #          return redirect('/')
class Login (View):  def get(self,request):  return render(request, 'bookmaking/login.html',
{'errors':'', 'username':''})  def post(self, request):  username1 = request.POST['username']
password = request.POST['password']  errors = []  user = authenticate(username=username1,
password=password)  if user is not None:  login(request, user)  return redirect('/')
errors.append('Логин или пароль неверны')  return render(request, 'bookmaking/login.html',
{'errors': mark_safe('<br>'.join(errors)), 'username': username1})  def log_in(request):  if
request.method == 'POST':  form = LoginModel(request.POST)  if form.is_valid():  if
form.get_user():  login(request, form.get_user())  return redirect('/')  else:
form=LoginModel()  return render(request, 'bookmaking/login2.html', {'form': form.as_p()})

```

models.py

```

from django.db import models from django import forms from django.contrib.auth.models import User
import pymysql from django.contrib.auth import authenticate pymysql.install_as_MySQLdb()

```

```

class User1(models.Model):  user_id = models.IntegerField()  name =
models.CharField(max_length=50)  age = models.IntegerField()  stake = models.FloatField()
contacts = models.CharField(max_length=50)  bank_account = models.CharField(max_length=50)
class Horse(models.Model):  horse_id = models.IntegerField(primary_key=True)  name =
models.CharField(max_length=50)  age = models.IntegerField()  ratio = models.FloatField()

```

```

class RegModel(forms.Form):  username1 = forms.CharField(label='Username', min_length=5)
password = forms.CharField(label='Password', min_length=8, widget=forms.PasswordInput)
password2 = forms.CharField(label='Repeat password', min_length=8, widget=forms.PasswordInput)
email = forms.CharField(label='Email', min_length=1)  surname = forms.CharField(label='Surname',
min_length=1)  name = forms.CharField(label='Name', min_length=1)  def clean (self):  value =
super(RegModel, self).clean()  password = value.get('password')  password2 =
value.get('password2')  if password!=password2:  raise forms.ValidationError('Пароли
должны совпадать')  users = User.objects.filter(username=value.get('username1'))  if
len(users)>0:  raise forms.ValidationError('Пользователь с таким именем уже существует')
class LoginModel(forms.Form):  username1 = forms.CharField(label='Username', min_length=5)
password = forms.CharField(label='Password', min_length=8, widget=forms.PasswordInput)  def
clean(self):  cleaned_data=super(LoginModel, self).clean()

```

```

        if not self.errors:  user = authenticate(username = cleaned_data['username1'], password =
cleaned_data['password'])  if user is None:  raise forms.ValidationError('Неверный логин
или пароль')  self.user=user  return cleaned_data  def get_user(self):  return self.user or
None

```

```
admin.py from django.contrib import admin from .models import User1
```

```
class MyAdmin(admin.ModelAdmin):    list_display = ('name', 'age', 'stake', 'contacts', 'bank_account')  
list_filter = ('stake',)    search_fields = ['name']    admin.site.register(User1, MyAdmin)
```

login.html

```
{% extends 'bookmaking/base.html' %} {% load staticfiles %}
```

```
{% block content %} {{ errors }}
```

```
<form action="/login" method="POST">  <label for="login">Логин</label>  <input type="text"  
name="username" id="login" value="{{ username }}"><br>
```

```
    <label for="password">Пароль</label> <input type="password" name="password"  
id="password"><br>
```

```
{% csrf_token %}
```

```
<button type="submit">Войти</button> </form> {% endblock content %}
```

Login2.html

```
{% extends 'bookmaking/base.html' %} {% load staticfiles %} {{ errors }}
```

```
{% block content %}
```

```
<form action="/login" method="POST">  {{ form }}
```

```
{% csrf_token %}
```

```
<button type="submit">Войти</button> </form> {% endblock content %}
```

Registration.html

```
{% extends 'bookmaking/base.html' %} {% load staticfiles %}
```

```
{% block content %} {{ errors }}
```

```
<form action="/registration" method="POST">  <label for="login">Логин</label>  <input type="text"  
name="username" id="login" value="{{ username }}"><br>
```

```
    <label for="password">Пароль</label>  <input type="password" name="password"  
id="password"><br>
```

```
    <label for="password">Пароль</label>  <input type="password" name="password2"  
id="password"><br>
```

```
    <label for="password">Email</label>  <input type="text" name="email" id="password" value="{{  
email }}"><br>
```

```
    <label for="password">Фамилия</label>  <input type="text" name="surname" id="password"  
value="{{ surname }}"><br>
```

```
    <label for="password">Имя</label>  <input type="text" name="name" id="password" value="{{  
name }}"><br>
```

{% csrf_token %}

<button type="submit">Зарегистрироваться</button> </form> {% endblock content %}

Registration2.html {% extends 'bookmaking/base.html' %} {% load staticfiles %} {{ errors }}

{% block content %}

<form action="/registration" method="POST"> {{ form }}

{% csrf_token %}

<button type="submit">Зарегистрироваться</button> </form> {% endblock content %}