

Защищено:
Гапанюк Ю.Е.

Демонстрация ЛР:
Гапанюк Ю.Е.

"__" _____ 2016 г.

"__" _____ 2016 г.

Отчет по лабораторной работе № 7 по курсу

Разработка интернет приложений

"Авторизация, работа с формами и Django Admin."

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53

(подпись)

Бирюкова Д. К.

"__" _____ 2016 г.

Задание лабораторной работы

Основная цель данной лабораторной работы – научиться обрабатывать веб-формы на стороне приложения, освоить инструменты, которые предоставляет Django, по работе с формами. Также освоить инструменты Django по работе с авторизацией и реализовать простейшую авторизацию. Также познакомиться с инструментом администрирования Django – как в несколько строчек кода сделать панель администратора сайта.

1. Создать view, которая возвращает форму для регистрации. Поля формы:

- Логин
- Пароль
- Повторный ввод пароля
- Email
- Фамилия
- Имя

2. Создать view, которая возвращает форму для авторизации. Поля формы:

- Логин
- Пароль

3. При отправке формы регистрации во view проверять каждый параметр по правилам валидации, если валидация всех полей пройдена, то создавать пользователя и делать перенаправление на страницу логина, а ошибки, если они есть, выводить над формой. Правила валидации:

- Логин не меньше 5 символов
- Пароль не меньше 8 символов
- Пароли должны совпадать
- Все поля должны быть заполнены
- Логин – уникален для каждого пользователя

4. При возникновении ошибок в момент отправки формы, введенные значения в полях ввода, кроме пароля, не должны исчезать.

5. Переписать view регистрации с использованием Django Form, правила валидации удалить из view, использовать встроенный механизм

валидации полей. 6. Во view авторизации реализовать логин при POST запросе. При успешной авторизации должен происходить переход на страницу успешной авторизации.

7. Страница успешной авторизации должна проверять, что пользователь авторизован. Иначе делать перенаправление на страницу авторизации.

8. Реализовать view для выхода из аккаунта.

9. Заменить проверку на авторизацию на декоратор login_required

10. Добавить superuser'a через команду manage.py

11. Подключить django.contrib.admin и войти в панель администрирования.

12. Зарегистрировать все свои модели в django.contrib.admin

13. Для выбранной модели настроить страницу администрирования:

- Настроить вывод необходимых полей в списке
- Добавить фильтры
- Добавить поиск
- Добавить дополнительное поле в список

ЛИСТИНГ

Файл urls.py:

```
from django.conf.urls import url
from django.contrib import admin
from lab7dj.views import *
urlpatterns = [
    url(r'^admin/', admin.site.urls), url(r'^registration/
    $', registration, name='registration'),
    url(r'^registration/success/$', registration_success),
    url(r'^authorization/$', authorization, name='authorization'),
    url(r'^authorization/success/$', auth_success), url(r'^exit/
    $', exit, name='exit'),
]
```

Файл admin.py:

```
from django.contrib import admin
from .models import *
# Register your models here. class
AnteAdmin(admin.ModelAdmin):
    list_display = ('id', 'amount')
    search_fields = ('id', 'amount')
admin.site.register(ante_model, AnteAdmin)
class TeamAdmin(admin.ModelAdmin):
    def win_pro(self, obj):
        p=obj.quantity_win/(obj.quantity_win+obj.quantity_lose
        ) return '%.2f' % p
    list_display = ('id', 'name',
'kind_of_sport', 'rating', 'quantity_win', 'quantity_lose', 'win_pro')
    list_filter = ['rating']
    search_fields = ('id', 'name')
admin.site.register(team_model, TeamAdmin)
class UserAdmin(admin.ModelAdmin):
    list_display = ('username', 'first_name',
'last_name') admin.site.register(User, UserAdmin)
```

Файл models.py:

```
from django.db import models
from django.contrib.auth.models import User
from django.contrib.auth.models import AbstractUser
# Create your models here.
class User(AbstractUser):
    age = models.IntegerField(default= 0 )
    phone = models.CharField(max_length=15, default='')
    is_in_black_list = models.BooleanField(default= 0)
    class Meta(AbstractUser.Meta):
        swappable = 'AUTH_USER_MODEL'
class ante_model (models.Model):
    amount = models.FloatField()
    user_for = models.ForeignKey(User, null=True) team_for
    = models.ForeignKey("team_model", null=True) def
__str__(self):
    return str(self.amount)
class team_model (models.Model):
    name = models.CharField(max_length=255, unique=True)
    kind_of_sport = models.CharField(max_length=255)
    rating = models.IntegerField(default=0) quantity_win
    = models.IntegerField(default=0) quantity_lose =
    models.IntegerField(default=0)
    def __str__(self): return
    str(self.name)
```

Файл views.py:

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from lab7dj.models import *
from django import forms
from django.contrib.auth.hashers import make_password
from django.contrib.auth import authenticate, logout
from django.contrib import auth
# Create your views here.
class RegistrationForm(forms.Form):
    username = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'username',
'placeholder': 'Введите логин', }), \
        min_length=5, label='Login:')
    name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'name',
'placeholder': 'Введите имя', }), \
        max_length=30, label='Name:')
    surname = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'surname',
'placeholder': 'Введите фамилию', }), \
        max_length=30, label='Surname:')
    email = forms.EmailField(
        widget=forms.EmailInput(attrs={'class': 'form-control', 'id': 'email',
'placeholder': 'Введите email', })
    )
    password = forms.CharField(min_length=8, label='Password:',
widget=forms.PasswordInput(
    attrs={'class': 'form-control', 'id': 'password', 'placeholder': 'Введите
пароль', }))
    password2 = forms.CharField(min_length=8, label='Confirm password:',
widget=forms.PasswordInput(
    attrs={'class': 'form-control', 'id': 'password2', 'placeholder': 'Повторите
пароль', }))
    def save(self):
        u = User()
        u.username = self.cleaned_data.get('username')
        u.password = make_password(self.cleaned_data.get('password'))
        u.first_name = self.cleaned_data.get('name')
        u.last_name = self.cleaned_data.get('surname')
        u.email = self.cleaned_data.get('email')
        u.is_staff = False
        u.is_active = True
        u.is_superuser = False
        u.save()
    def clean_password2(self):
        if self.cleaned_data.get('password') != self.cleaned_data.get('password2'):
            raise forms.ValidationError('Passwords does not match')
    def clean_username(self):
        username = self.cleaned_data.get('username')
        try:
            u = User.objects.get(username=username)
            raise forms.ValidationError('This login already uses')
        except User.DoesNotExist:
            return username
#####
def registration(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('success/')
        return render(request, 'registration.html', {'form':
form})
    else:
        form = RegistrationForm()
    return render(request, 'registration.html', {'form': form})
```

```

def registration_success (request):
    return render(request, 'reg_success.html')
def authorization (request):
    errors=[]
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        if not username:
            errors.append(' Введите логин ')
        elif not password:
            errors.append(' Введите пароль ')
        else:
            user = authenticate(username=username, password=password)
            if user:
                auth.login(request, user)
                return redirect('success/')
            else:
                errors.append('Неправильный логин или пароль')
    return render(request, 'authorization.html', {'errors': errors, 'username':
username})
    return render(request, 'authorization.html', {'users': errors})
@login_required(login_url="/authorization/")
def auth_success(request):
    return render(request, 'auth_success.html')
@login_required
def exit(request):
    logout(request)
    return render(request, 'exit.html')

```

Файл settings.py: (изменения)

```

DATABASES = {
    'default': {
        'ENGINE' : 'django.db.backends.mysql',
        'NAME': "bets_db",
        'USER': 'olga_user',
        'PASSWORD': "123",
        'HOST': "localhost",
        'PORT': 3306, # Стандартный порт MySQL
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

```

AUTH_USER_MODEL = 'lab7dj.User'

Файл auth_success.html:

```

{% extends 'base_page.html' %}
{% block title %}
    Lab_7_RIP {%
endblock %} {%
block head %}
    Поздравляем!Авторизация прошла успешно!
{% endblock %}
{% block body %}
    Вы авторизованы.
    <a href= "{% url 'exit' %}" class="btn btn-
primary">Выйти</a> {% endblock %}

```

Файл authorization.html:

```

{% extends 'base_page.html' %}
{% block title %}
    Lab_7_RIP {%
endblock %} {%
block head %}
    Авторизация

```

```
{% endblock %}
{% block body %}
<form class="form-horizontal" action="" method="POST">
  {% csrf_token %}
  <div class="form-group" >
    {% for error in errors %}
      <div class="alert alert-danger" role="alert">
        <strong>{{ error }}</strong>
      </div>
    {% endfor %}
  </div>
  <div class="form-group">
    <label for="id_username" class="control-label col-sm-2">Логин:</label> <div class="col-sm-3">
      <input class="form-control" type="text" name="username" id="id_username"
value='{{ username }}'>
    </div>
  </div>
  <div class="form-group">
    <label for="id_password" class="control-label col-sm-2">Пароль:</label>
    <div class="col-sm-3">
      <input class="form-control" type="password" name="password" id="id_password">
    </div>
  </div>
  <div class="form-group">
    <div class="col-xs-offset-2 col-xs-1">
      <button type="submit" class="btn btn-primary" style="float:
left">Войти</button>
    </div>
    <div class="col-xs-2">
      <a href="{% url 'registration' %}" class="btn btn-default" style="float:
right">Зарегистрироваться</a>
    </div>
  </div>
</div>
</form>
>
{% endblock %}
```

Файл base_page.html:

```
{% load static %}
<!DOCTYPE html>
<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge"> <title>Lab_7_RIP</title>
  <link href="{% static "my_lab7/css/bootstrap.min.css" %}" rel="stylesheet">
  <link href="/static/my_lab7/css/jumbotron.css" rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top"> <div class="container">
    <p class="navbar-brand" >{% block title %}{% endblock %}</p>
  </div>
</nav>
<div class="jumbotron">
  <div class="container">
    <h1>{% block head %}{% endblock %}</h1>
  </div>
</div>
<div class="container">
  {% block body %}{% endblock %}
</div>
<hr> </hr>
</body></html>
```

Файл exit.html:

```
{% extends 'base_page.html' %}
{% block title %}
    Lab_7_RIP {%
endblock %} {%
block head %}
    Выход!
{% endblock %}
{% block body %}
    <p>Вы вышли из аккаунта.</p>

{% endblock %}
```

Файл reg_success.html:

```
{% extends 'base_page.html' %}
{% block title %}
    Lab_7_RIP {%
endblock %} {%
block head %}
    Поздравляем!Регистрация прошла успешно!
{% endblock %}
{% block body %}
    Вы зарегистрированы.
{% endblock %}
```

Файл registration.html:

```
{% extends 'base_page.html' %}
{% block title %}
    Lab_7_RIP {%
endblock %} {%
block head %}
    Регистрация
{% endblock %}
{% block body %}
    <form class="form-horizontal" method="POST">
    {% csrf_token %}
    <div class="form-group">
        {% if form.errors %}
            <div class="col-md-4 col-md-offset-2"> <div class="alert alert-warning">
                <a href= "/reg" class="close" data-dismiss="alert"></a> {% for f in form %}
                    {% if f.errors %} <p>{{
                        f.label }}</p> {% for e
                        in f.errors %}
                            <p>{{ e }}</p>
                        {% endfor %}
                    {% endif %}
                {% endfor %}
            </div>
        </div>
        {% endif %}
    </div>
    <div class="form-group">
        <label class="col-xs-2 control-label">ЛОГИН:</label>
        <div class="col-xs-4">
            <div class="input-group">
                {{ form.username }}
            </div>
        </div>
    </div>
    <div class="form-group">
        <label class="col-xs-2 control-label">Имя:</label> <div class="col-xs-4">
            <div class="input-group">
                {{ form.name }}
```

```

        </div>
    > </div>
</div>
<div class="form-group">
    <label class="col-xs-2 control-label">Фамилия:</label>
    <div class="col-xs-4">
        <div class="input-group">
            {{ form.surname }}
        </div>
    > </div>
</div>
<div class="form-group">
    <label class="col-xs-2 control-label">Email:</label>
    <div class="col-xs-4">
        <div class="input-group">
            {{ form.email }}
        </div>
    > </div>
</div>
<div class="form-group">
    <label for="inputPassword" class="col-xs-2 control-label">Пароль:</label> <div class="col-xs-4">
        <div class="input-group">
            {{ form.password }}
        </div>
    > </div>
</div>
<div class="form-group">
    <label for="confirmPassword" class="col-xs-2 control-label">Повтор
пароля:</label>
    <div class="col-xs-4">
        <div class="input-group">
            {{ form.password2 }}
        </div>
    > </div>
</div>
<div class="form-group">
    <div class="col-xs-offset-2 col-xs-1">
        <button type="submit" class="btn btn-primary" style="float:
left">Зарегистрироваться</button>
    </div>
    <div class="col-xs-2">
        <a href="{% url 'authorization' %}" class="btn btn-default"
style="float: right">Войти</a>
    </div>
> </div>
</form>
{% endblock %}

```


Результаты работы программы

