МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА

Кафедра ИУ-5

«Системы обработки информации и управления»

Отчет по лабораторной работе № 7

По предмету «Разработка интернет-приложений»

Выполнил: Лузин Д. С.

Группа: ИУ5-53

**1. Задание**

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.

1. Создайте view, которая возвращает форму для регистрации
2. Создайте view, которая возвращает форму для авторизации
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
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 комманду manage.py
11. Подключить django.contrib.admin и войти в панель администрирования.
12. Зарегистрировать все свои модели в django.contrib.admin

**2. Листинги файлов проекта**

**Файл forms.py:**

**from** django.contrib.auth **import** authenticate  
**from** django.contrib.auth.hashers **import** make\_password  
**from** django **import** forms  
**from** django.contrib.auth.models **import** User  
  
  
**class** RegistrationForm(forms.Form):  
 username = forms.CharField(  
 widget=forms.TextInput(attrs={**'class'**: **'form-control'**, **'placeholder'**: **'Username'**, }),  
 max\_length=30, min\_length=5, required=**True**, label=**'Login'** )  
 first\_name = forms.CharField(  
 widget=forms.TextInput(attrs={**'class'**: **'form-control'**, **'placeholder'**: **u'Имя'**, }),  
 max\_length=30, required=**True**, label=**'Firstname'** )  
 last\_name = forms.CharField(  
 widget=forms.TextInput(attrs={**'class'**: **'form-control'**, **'placeholder'**: **u'Фамилия'**, }),  
 max\_length=30, required=**True**, label=**'Lastname'** )  
 email = forms.EmailField(  
 widget=forms.TextInput(attrs={**'class'**: **'form-control'**, **'placeholder'**: **'E-mail'**, }),  
 required=**True**, max\_length=254, label=**'E-mail'** )  
 password1 = forms.CharField(  
 widget=forms.PasswordInput(attrs={**'class'**: **'form-control'**, **'placeholder'**: **'Password'**}),  
 min\_length=8, label=**'Password'** )  
 password2 = forms.CharField(  
 widget=forms.PasswordInput(attrs={**'class'**: **'form-control'**, **'placeholder'**: **'Password'**}),  
 min\_length=8, label=**u'Repeat password'** )  
  
 **def** clean\_username(self):  
 username = self.cleaned\_data.get(**'username'**, **''**)  
  
 **try**:  
 u = User.objects.get(username=username)  
 **raise** forms.ValidationError(**'Such user already exists'**)  
 **except** User.DoesNotExist:  
 **return** username  
  
 **def** clean\_password2(self):  
 pass1 = self.cleaned\_data.get(**'password1'**, **''**)  
 pass2 = self.cleaned\_data.get(**'password2'**, **''**)  
  
 **if** pass1 != pass2:  
 **raise** forms.ValidationError(**'Passwords do not match'**)  
  
 **def** save(self):  
 data = self.cleaned\_data  
 password = data.get(**'password1'**)  
 u = User()  
  
 u.username = data.get(**'username'**)  
 u.password = make\_password(password)  
 u.email = data.get(**'email'**)  
 u.first\_name = data.get(**'first\_name'**)  
 u.last\_name = data.get(**'last\_name'**)  
 u.is\_active = **True** u.is\_superuser = **False** u.save()  
  
 **return** authenticate(username=u.username, password=password)  
  
**class** LoginForm(forms.Form):  
 login = forms.CharField(  
 widget=forms.TextInput(  
 attrs={**'class'**: **'form-control'**, **'placeholder'**: **'Username'**, }  
 ),  
 max\_length=30,  
 label=**'Login'** )  
 password = forms.CharField(  
 widget=forms.PasswordInput(  
 attrs={**'class'**: **'form-control'**, **'placeholder'**: **'Password'**, }  
 ),  
 label=**'Passsword'** )  
  
 **def** clean(self):  
 data = self.cleaned\_data  
 user = authenticate(username=data.get(**'login'**, **''**), password=data.get(**'password'**, **''**))  
  
 **if** user **is not None**:  
 **if** user.is\_active:  
 data[**'user'**] = user  
 **else**:  
 **raise** forms.ValidationError(**'not active'**)  
 **else**:  
 **raise** forms.ValidationError(**'Wrong username or password'**)

**Файл views.py:**

**from** django.shortcuts **import** render  
**from** django.contrib **import** auth  
**from** django.http.response **import** HttpResponseRedirect  
**from** django.views **import** View  
**from** login.models **import** \*  
**from** login.forms **import** \*  
  
**def** registration(request):  
  
 **if** request.user.is\_authenticated():  
 **return** render(request, **'loggedin.html'**)  
  
 **if** request.method == **"POST"**:  
 form = RegistrationForm(request.POST)  
 **if** form.is\_valid():  
 user = form.save()  
 auth.login(request, user)  
 **return** render(request, **'registered.html'**)  
 **else**:  
 form = RegistrationForm()  
  
 **return** render(request, **'registration.html'**, {  
 **'form'**: form  
 })  
  
  
**def** login(request):  
 **if** request.method == **"POST"**:  
 form = LoginForm(request.POST)  
  
 **if** form.is\_valid():  
 auth.login(request, form.cleaned\_data[**'user'**])  
 **return** render(request, **'loggedin.html'**)  
 **else**:  
 form = LoginForm()  
 **return** render(request, **'login.html'**, {  
 **'form'**: form  
 })  
  
**def** loggedin(request):  
 **return** render(request, **'loggedin.html'**)

**Форма регистрации**



