Московский государственный технический университет им. Н.Э.Баумана

Проверил:
Гапанюк Ю.Е.

"__"_____2019 г.

Отчет по лабораторной работе № 6 по курсу “Разработка интернет-приложений”

« Работа с формами, авторизация и модуль администрирования в Django»

7
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-54** _____

(подпись)

Меркулова Н. А.

"__"_____2019 г.

Москва, МГТУ - 2019

1. 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.

1. Создайте view, которая возвращает форму для регистрации.

Поля формы:
- Логин
- Пароль
- Повторный ввод пароля
- Email
- Фамилия
- Имя

2. Создайте 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'а через комманду 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

2. Описание проекта

В качестве темы проекта был выбран форум с вопросами (техническое задание взято отсюда https://github.com/ziontab/tp-tasks/blob/master/files/markdown/technical_details.md).

3. Исходный код

3.1. Форма логина

```python
class LoginForm(forms.Form):
    login = forms.CharField(
        label='Login',
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter Username '}),
        max_length=30
    )

    password = forms.CharField(
        label='Password',
        widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': '*******'}),
        min_length=8
    )

    def clean(self):
        data = self.cleaned_data
        user_username = data.get('login', '')
        user_password = data.get('password', '')

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

        if user is not None:
            if user.is_active:
                data['user'] = user
            else:
                raise forms.ValidationError('User is not active')
        else:
            raise forms.ValidationError('Wrong login or password')
```

3.2. Форма регистрации

```
class SignupForm(forms.Form):
    username = forms.CharField(
        label='Login',
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'login'}),
        max_length=30
    )

    email = forms.EmailField(
        label='Email',
        widget=forms.EmailInput(attrs={'class': 'form-control', 'placeholder': 'e@mail.ru'}),
        max_length=100
    )

    nick_name = forms.CharField(
        label='NickName',
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'NickName'}),
        max_length=30
    )

    password = forms.CharField(
        label='Password',
        widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': '*******'}),
        min_length=8
    )

    password_repeat = forms.CharField(
        label='Repeat Password',
        widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': '*******'}),
        min_length=8
    )

    avatar = forms.FileField(
        label='Upload avatar',
        widget=forms.ClearableFileInput(attrs={}),
        required=False
    )

    def clean_username(self):
        username = self.cleaned_data.get('username', '')

        try:
            User.objects.get(username=username)

            raise forms.ValidationError('Username is already used')
        except User.DoesNotExist:

            return username
```

```
def clean_nick_name(self):
    nickname = self.cleaned_data.get('nick_name', '')

    try:
        User.objects.get(last_name=nickname)

        raise forms.ValidationError('NickName is already used')
    except User.DoesNotExist:

        return nickname

def clean_password_repeat(self):
    password = self.cleaned_data.get('password', '')
    password_repeat = self.cleaned_data.get('password_repeat', '')

    if password != password_repeat:
        raise forms.ValidationError('Passwords does not matched')

    return password_repeat

def clean_email(self):
    email = self.cleaned_data.get('email', '')

    try:
        User.objects.get(email=email)

        raise forms.ValidationError('Email is already used')
    except User.DoesNotExist:
        return email

def clean_avatar(self):
    avatar = self.cleaned_data.get('avatar')

    return avatar
```

```
def save(self):
    data = self.cleaned_data
    password = data.get('password')
    u = User()

    u.username = data.get('username')
    u.last_name = data.get('nick_name')
    u.password = make_password(password)
    u.email = data.get('email')
    u.is_active = True
    u.is_superuser = False
    u.save()

    up = Profile()
    up.user = u
    up.rating = 0

    if data.get('avatar') is not None:
        avatar = data.get('avatar')
        up.avatar.save('%s.png' % u.username, avatar, save=True)

    up.save()

    return authenticate(username=u.username, password=password)
```

3.3. View логина и логаута

```
def login(request):
    redirect = request.GET.get('continue', '/')
    if request.user.is_authenticated:
        return HttpResponseRedirect(redirect)
    if request.method == "POST":
        form = LoginForm(request.POST)
        if form.is_valid():
            auth.login(request, form.cleaned_data['user'])
            return HttpResponseRedirect(redirect)
    else:
        form = LoginForm()

    return render(request, 'login.html', {
        'form': form,
        'popular_tags': model_manager.popular_tags(request),
        'best_members': model_manager.best_members(request)
    })


@login_required
def logout(request):
    redirect = request.GET.get('continue', '/')
    auth.logout(request)

    return HttpResponseRedirect(redirect)
```

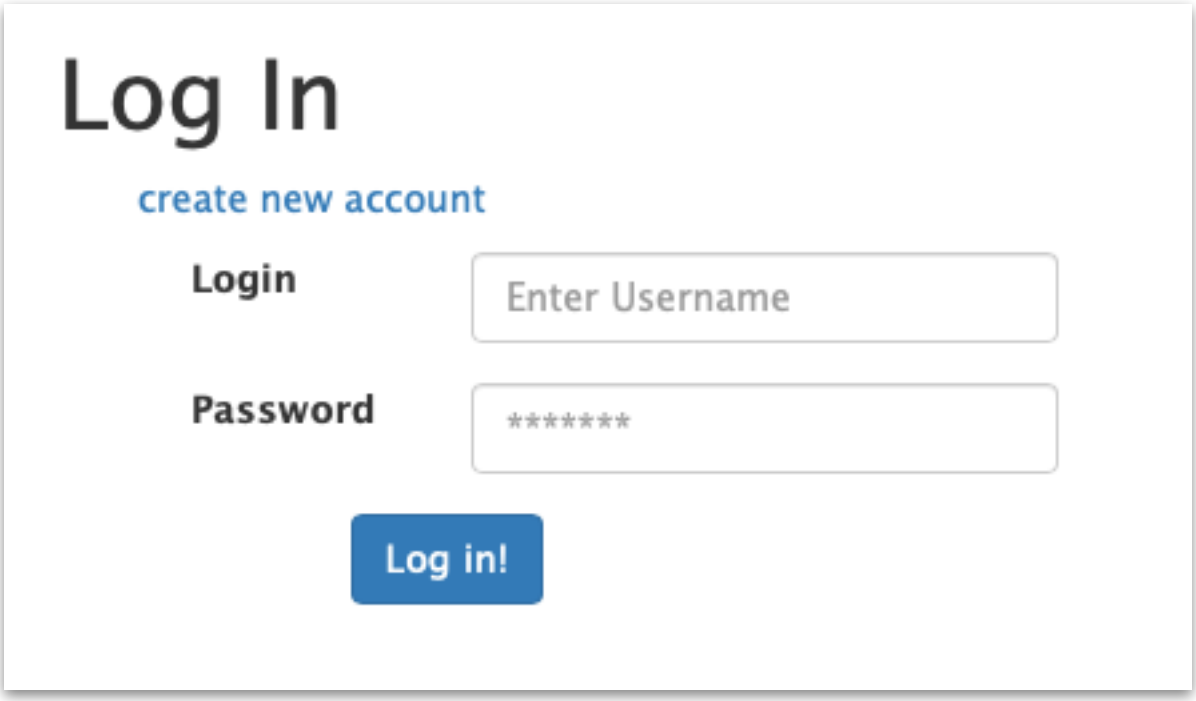
3.4. View регистрации

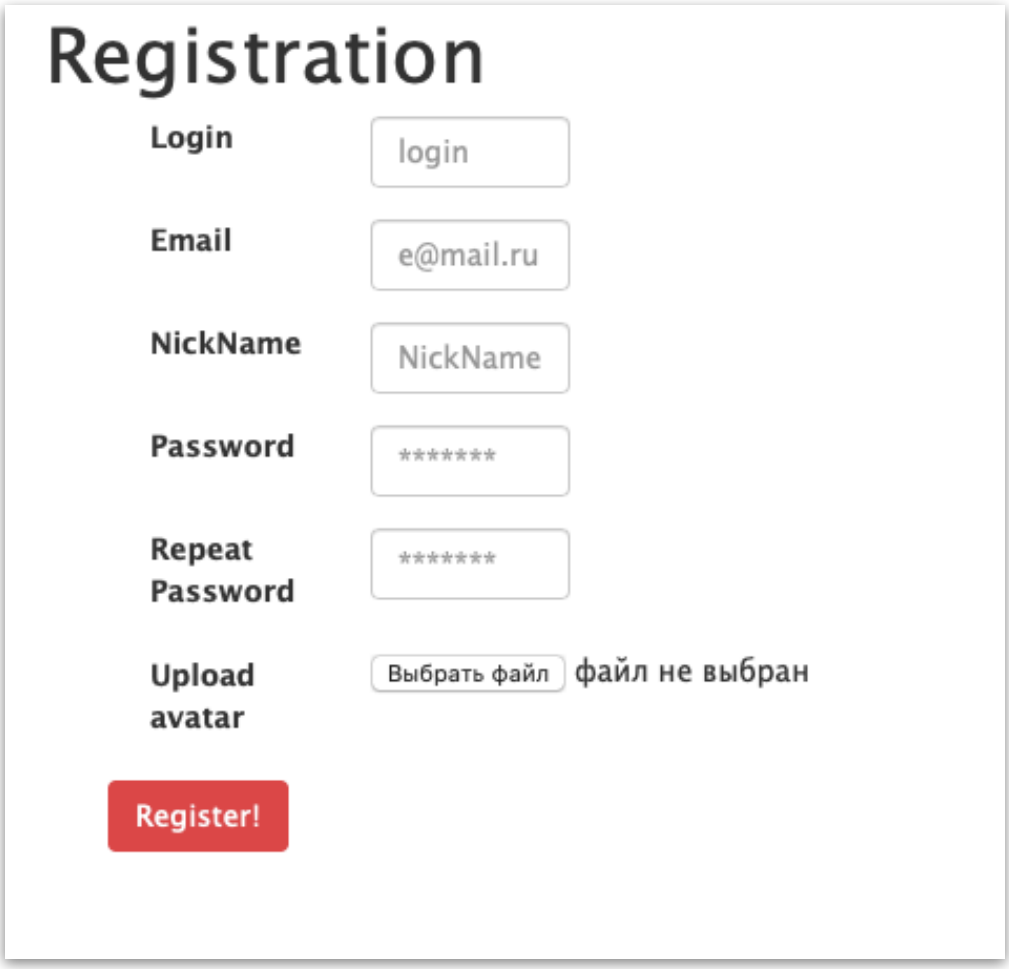
```
def signup(request):
    context = {
        'popular_tags': model_manager.popular_tags(request),
        'best_members': model_manager.best_members(request),
    }

    if request.user.is_authenticated:
        return HttpResponseRedirect('/')
    if request.method == "POST":
        form = SignupForm(request.POST, request.FILES)
        if form.is_valid():
            user = form.save()
            auth.login(request, user)
            return HttpResponseRedirect('/')
    else:
        form = SignupForm()

    return render(request, 'signup.html', {'form': form, 'popular_tags': model_manager.popular_tags(request),
                                           'best_members': model_manager.best_members(request)})
```

4. Скриншоты


Log In
create new account
Login
Enter Username
Password
****
Log in!


Registration
Login
login
Email
e@mail.ru
NickName
NickName
Password
****
Repeat Password
****
Upload avatar
Выбрать файл
файл не выбран
Register!