Московский государственный технический

университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №7

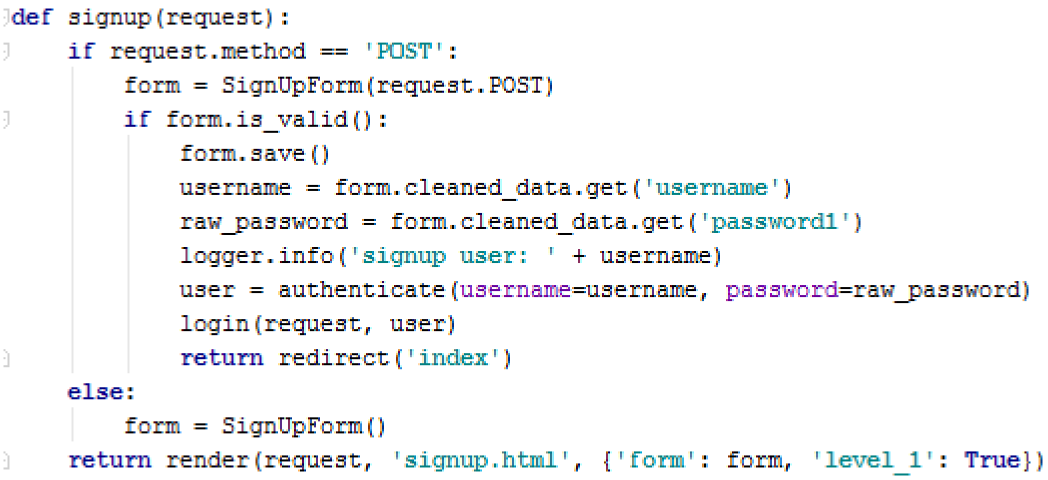
Выполнила:

студент группы ИУ5-52

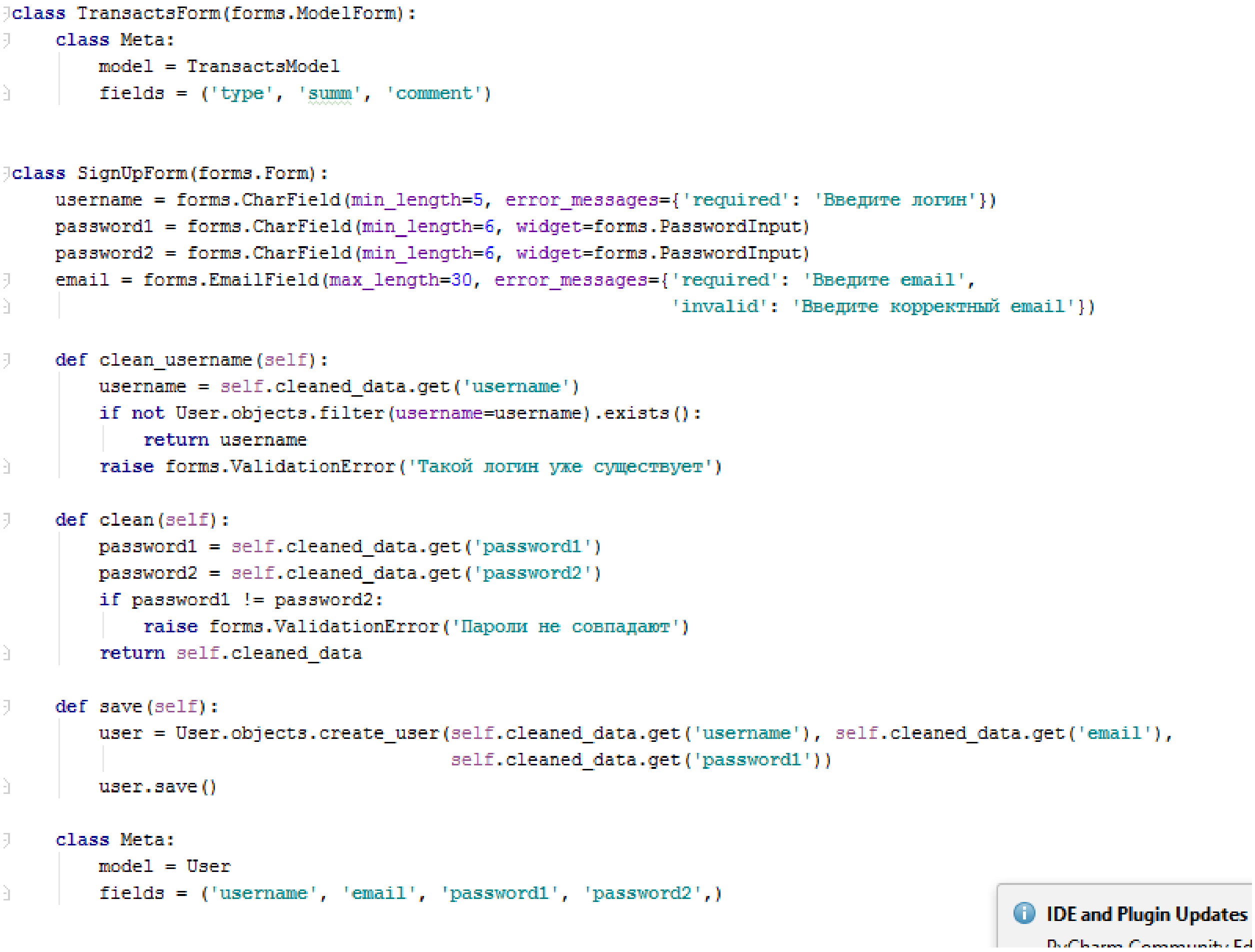
Кучаева К.И.

Москва, 2017 г

Форма регистрации

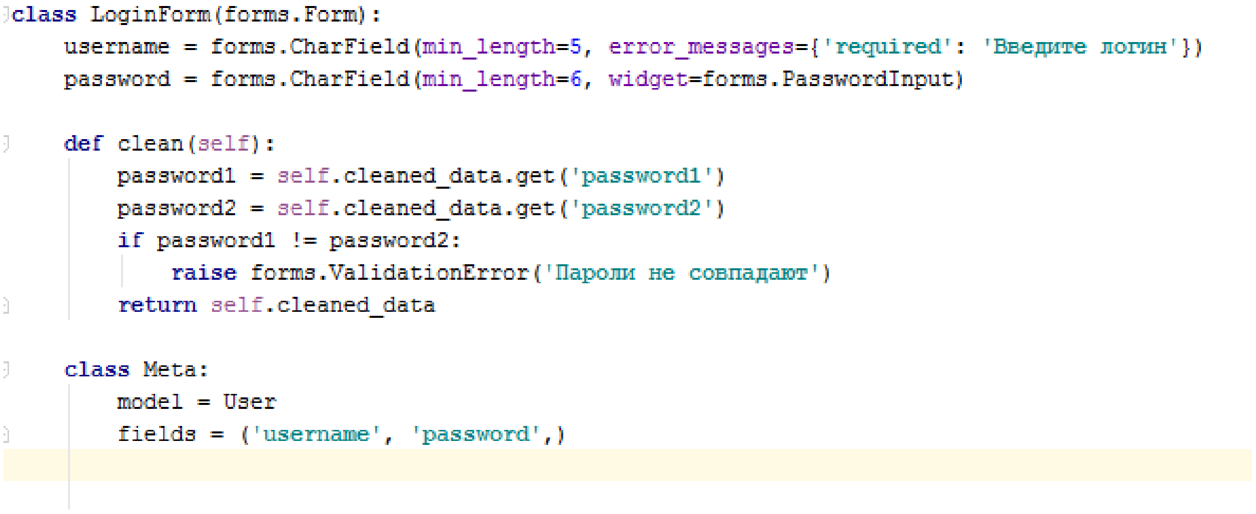


```
def signup(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            raw_password = form.cleaned_data.get('password1')
            logger.info('signup user: ' + username)
            user = authenticate(username=username, password=raw_password)
            login(request, user)
            return redirect('index')
    else:
        form = SignUpForm()
    return render(request, 'signup.html', {'form': form, 'level_1': True})
```

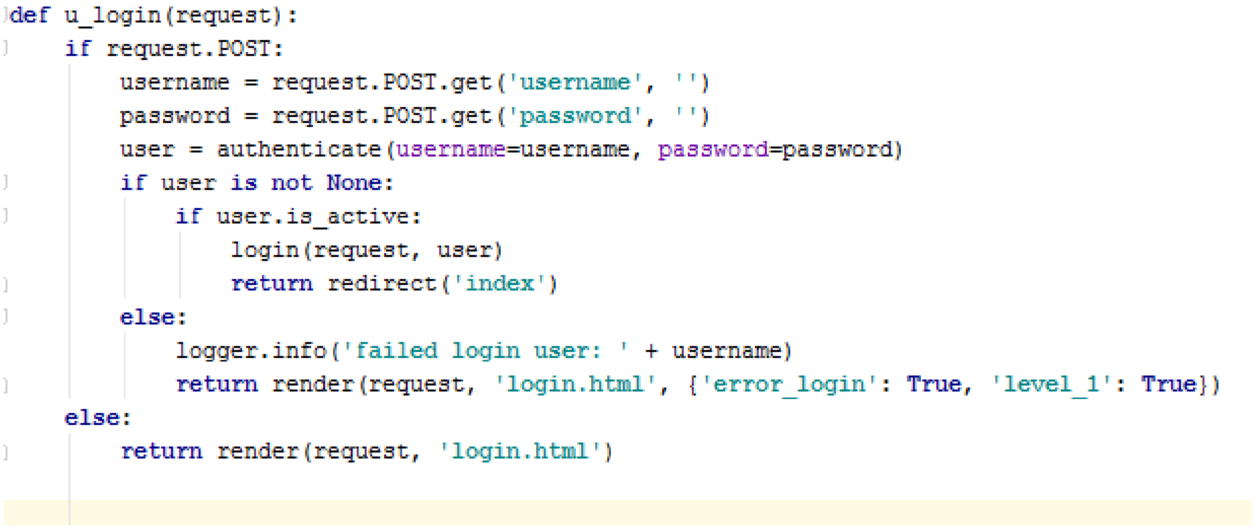


```
class TransactsForm(forms.ModelForm):
    class Meta:
        model = TransactsModel
        fields = ('type', 'summ', 'comment')


class SignUpForm(forms.Form):
    username = forms.CharField(min_length=5, error_messages={'required': 'Введите логин'})
    password1 = forms.CharField(min_length=6, widget=forms.PasswordInput)
    password2 = forms.CharField(min_length=6, widget=forms.PasswordInput)
    email = forms.EmailField(max_length=30, error_messages={'required': 'Введите email',
                                                            'invalid': 'Введите корректный email'})

    def clean_username(self):
        username = self.cleaned_data.get('username')
        if not User.objects.filter(username=username).exists():
            return username
        raise forms.ValidationError('Такой логин уже существует')

    def clean(self):
        password1 = self.cleaned_data.get('password1')
        password2 = self.cleaned_data.get('password2')
        if password1 != password2:
            raise forms.ValidationError('Пароли не совпадают')
        return self.cleaned_data

    def save(self):
        user = User.objects.create_user(self.cleaned_data.get('username'), self.cleaned_data.get('email'),
                                        self.cleaned_data.get('password1'))
        user.save()

    class Meta:
        model = User
        fields = ('username', 'email', 'password1', 'password2',)
```

Форма авторизации

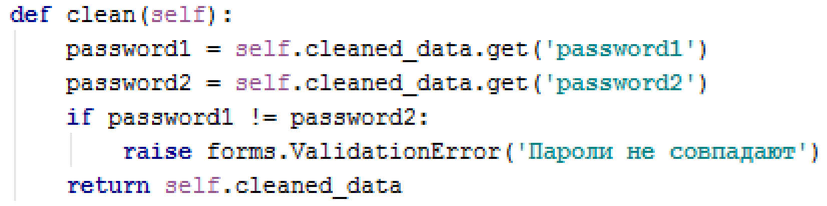


```
class LoginForm(forms.Form):
    username = forms.CharField(min_length=5, error_messages={'required': 'Введите логин'})
    password = forms.CharField(min_length=6, widget=forms.PasswordInput)

    def clean(self):
        password1 = self.cleaned_data.get('password1')
        password2 = self.cleaned_data.get('password2')
        if password1 != password2:
            raise forms.ValidationError('Пароли не совпадают')
        return self.cleaned_data

    class Meta:
        model = User
        fields = ('username', 'password',)
```



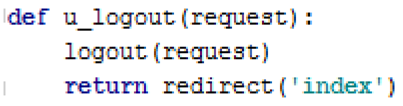
```
def u_login(request):
    if request.POST:
        username = request.POST.get('username', '')
        password = request.POST.get('password', '')
        user = authenticate(username=username, password=password)
        if user is not None:
            if user.is_active:
                login(request, user)
                return redirect('index')
        else:
            logger.info('failed login user: ' + username)
            return render(request, 'login.html', {'error_login': True, 'level_1': True})
    else:
        return render(request, 'login.html')
```

Валидация



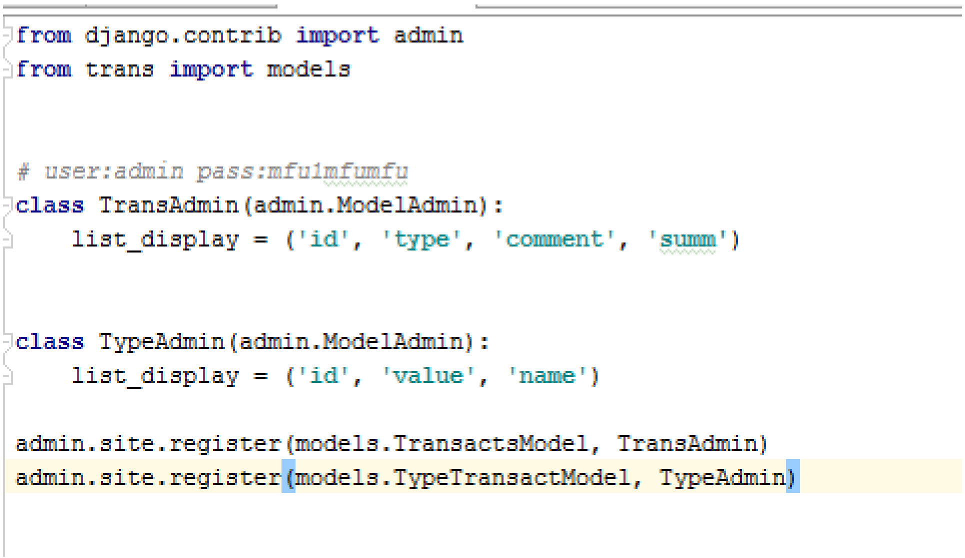
```
def clean(self):
    password1 = self.cleaned_data.get('password1')
    password2 = self.cleaned_data.get('password2')
    if password1 != password2:
        raise forms.ValidationError('Пароли не совпадают')
    return self.cleaned_data
```

Выход из аккаунта



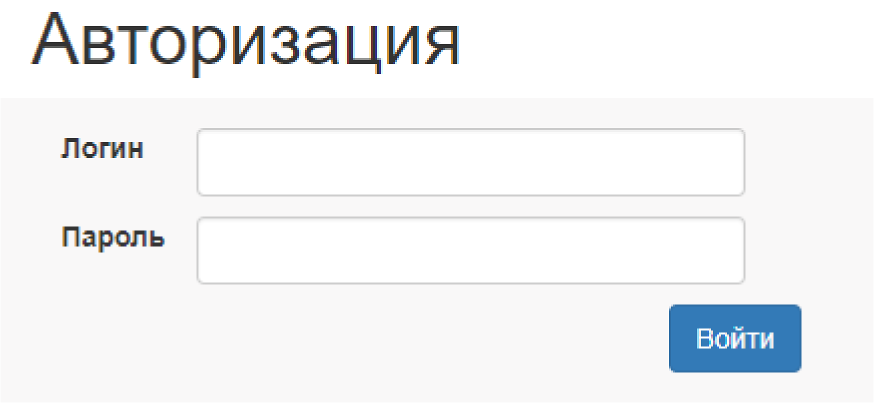
```
def u_logout(request):
    logout(request)
    return redirect('index')
```

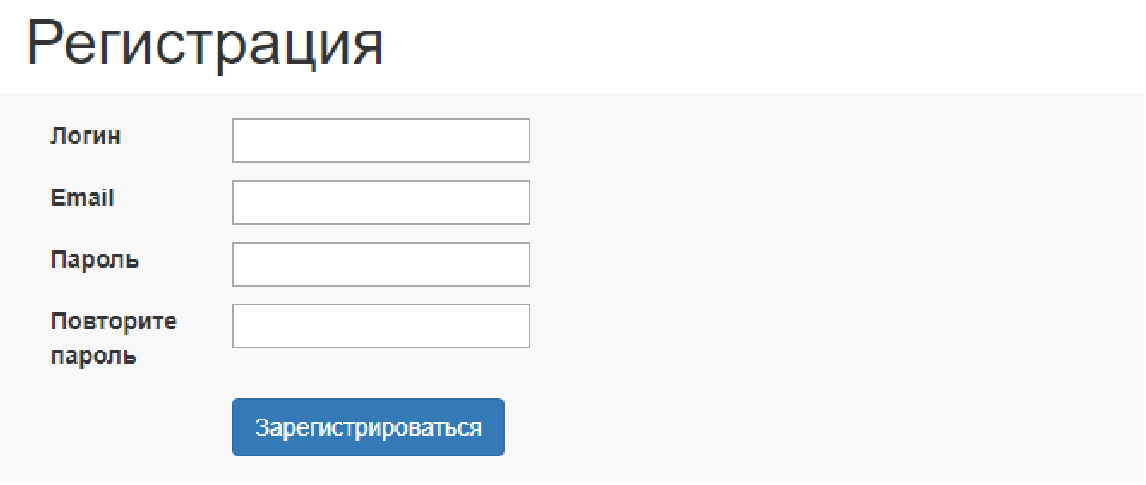
Django admin



```
from django.contrib import admin
from trans import models


# user:admin pass:mfu1mfumfu
class TransAdmin(admin.ModelAdmin):
    list_display = ('id', 'type', 'comment', 'summ')


class TypeAdmin(admin.ModelAdmin):
    list_display = ('id', 'value', 'name')

admin.site.register(models.TransactsModel, TransAdmin)
admin.site.register(models.TypeTransactModel, TypeAdmin)
```

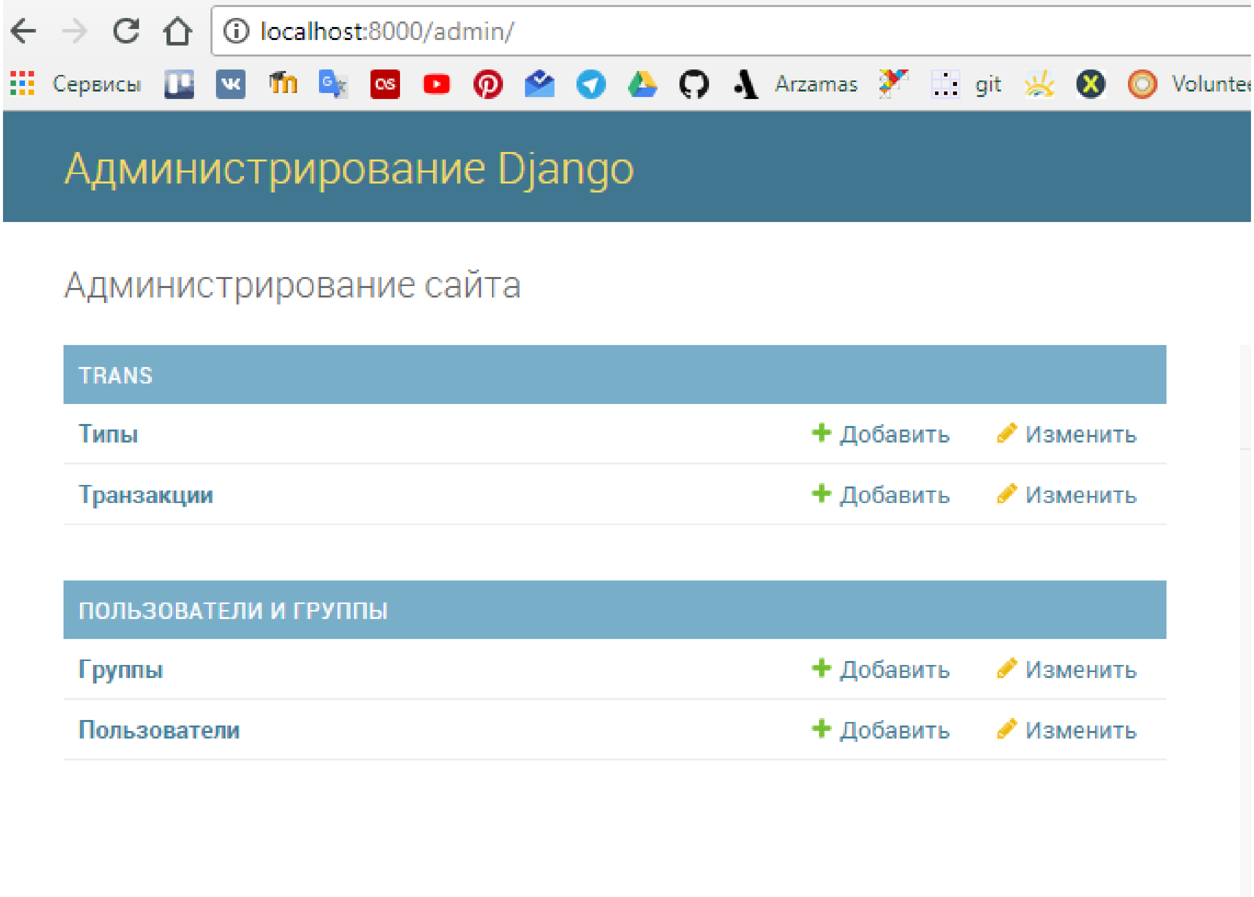
Результат





Регистрация

Логин	
Email	
Пароль	
Повторите пароль	



Администрирование Django

Администрирование сайта

TRANS		
Типы	+ Добавить	Изменить
Транзакции	+ Добавить	Изменить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ		
Группы	+ Добавить	Изменить
Пользователи	+ Добавить	Изменить