

**Московский государственный технический университет им.
Н.Э.Баумана**

Кафедра «Системы обработки информации и управления»

Разработка интернет приложений

Отчет по Лабораторной работе №7.

Исполнитель: студент группы РТ5-51

Бгатцев А. В.

Москва – 2016

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 Используемые формы

```
class LoginForm(forms.Form):
    login = forms.CharField(label='Логин', min_length=5)
    password = forms.CharField(label='Пароль', min_length=8,
                               widget=forms.PasswordInput)

class RegisterForm(forms.Form):
    login = forms.CharField(label='Логин', min_length=5)
    password = forms.CharField(label='Пароль', min_length=8,
                               widget=forms.PasswordInput)
    password2 = forms.CharField(label='Подтверждение пароля', min_length=8,
                                widget=forms.PasswordInput)
    email = forms.CharField(label='Email', min_length=1)
    firstname = forms.CharField(label='Имя', min_length=1)
    lastname = forms.CharField(label='Фамилия', min_length=1)

    def clean(self):
        cleaned_data = super(RegisterForm, self).clean()
        password = cleaned_data.get('password')
        password2 = cleaned_data.get('password2')
        if password != password2:
            raise forms.ValidationError("Пароли не совпадают")
        users = User.objects.filter(username=cleaned_data.get('login'))
        if len(users) > 0:
            raise forms.ValidationError("Пользователь с данным логином уже существует")
```

2 Используемые модели

```
class Catalog(models.Model):
    id = models.AutoField(auto_created=True, primary_key=True,
                          serialize=False)
    product = models.CharField(max_length=255)
    category = models.CharField(max_length=255)
    price = models.CharField(max_length=255)
    quantity = models.CharField(max_length=255)
    foto = models.CharField(max_length=255)

    def image_img(self):
        if self.foto:
            return u'<a href="{0}" target="_blank"></a>'.format(self.foto)
        else:
            return ' (Нет изображения) '
    image_img.short_description = 'Картинка'
    image_img.allow_tags = True
```

3 Калассы Вода и Регистрации во Views

```
class Register(View):
    def get(self, request):
        form = RegisterForm()
        return render(request, 'register.html', {'errors': '', 'form':
form.as_p()})

    def post(self, request):
        form = RegisterForm(request.POST)

        if not form.is_valid():
            return render(request, 'register.html', {'errors': '', 'form':
form.as_p()})

        u = User(username=form.cleaned_data['login'],
email=form.cleaned_data['email'],
                last_name=form.cleaned_data['lastname'],
first_name=form.cleaned_data['firstname'])
        u.set_password(form.cleaned_data['password'])
        u.save()
        return redirect('/order')

class Login(View):
    def get(self, request):
        form = LoginForm()
        return render(request, 'login.html', {'form': form.as_p()})

    def post(self, request):
        log = request.POST['login']
        password = request.POST['password']
        errors = []

        user = authenticate(username=log, password=password)

        if user is not None:
            login(request, user)
            return redirect('/order')
        errors.append('Логин или пароль неверны')
        return render(request, 'login.html', {'errors':
mark_safe('<br>'.join(errors)), 'login': login})

@login_required(login_url='/login')
def authOrder(request):
    a = 'Ваши заказы'
    return render(request, 'authOrder.html', {'auth': a})

class Logout(View):
    success_url = '/'
    def get(self, request):
        logout(request)
        return HttpResponseRedirect('/')

```

4 Настройка админки

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

# Register your models here.

class PostAdmin(admin.ModelAdmin):
    list_display = ('product', 'category', 'price', 'quantity', 'foto',
'image_img' )
    list_filter = ('category',)
    search_fields = ['product']
    readonly_fields = ['image_img', ]

admin.site.register(Catalog, PostAdmin)
```

5 Скриншоты



