

Демонстрация ЛР:  
Гапанюк Ю.Е.

**Отчет по лабораторной работе № 7 по курсу  
разработка интернет приложений  
"Авторизация, работа с формами и Django Admin"**

Вариант № <15>

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-54

Матвейчук И.А.

# Задание

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 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

# Текст программы

## Файл admin.py:

```
from django.contrib import admin
from .models import ComputerModel, OrderModel, User

class ComputerAdmin(admin.ModelAdmin):
    list_display = ('serial_num', 'brand')
    search_fields = ('serial_num', 'brand')

admin.site.register(ComputerModel, ComputerAdmin)

class OrderAdmin(admin.ModelAdmin):
    def item_amount(self, obj):
        data = ComputerModel.objects.filter(ord_num=obj.id).all()
        i = len(data)
        return i
    list_display = ('order_num', 'order_date', 'item_amount')
    list_filter = ['order_date']

admin.site.register(OrderModel, OrderAdmin)

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 AbstractUser

class ComputerModel(models.Model):
    id = models.AutoField(primary_key=True)
    serial_num = models.IntegerField(unique=True, verbose_name='Серийный номер')
    #comp_pic = models.ImageField()
    brand = models.CharField(max_length=20, default='Dell') # производитель
    type = models.CharField(max_length=10, default='laptop') # desktops, laptop,
tablet
    screen_size = models.FloatField(default=11)
    installed_OS = models.CharField(max_length=20, default='Windows8')
    processor_type = models.CharField(max_length=20, default='Inspiron 11')
    RAM = models.FloatField(default=2)
    price = models.CharField(max_length=20, default='30000 руб')
    ord_num = models.ManyToManyField('OrderModel')

    def __str__(self):
        return str(self.serial_num)

class User(AbstractUser):
    phone = models.CharField(max_length=15, default='')
    adress = models.TextField(verbose_name='Адрес', default='')

    class Meta(AbstractUser.Meta):
        swappable = 'AUTH_USER_MODEL'

class OrderModel(models.Model):
    id = models.AutoField(primary_key=True)
    order_num = models.IntegerField(unique=True)
```

```
order_date = models.DateField()
user_id = models.ForeignKey(User)
```

```
def __str__(self):
    return str(self.order_num)
```

### Файл forms.py:

```
from django import forms
from my_lab7_1.models import *
from django.contrib.auth.hashers import make_password

class RegistrationForm(forms.Form):
    username = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'username'}), \
                               min_length=5, label='Логин:')
    name = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'name'}), \
                           max_length=30, label='Имя:')
    surname = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'id': 'surname'}), \
                              max_length=30, label='Фамилия:')
    email = forms.EmailField(widget=forms.EmailInput(attrs={'class': 'form-control', 'id': 'email'}), label='Email')
    password = forms.CharField(widget=forms.PasswordInput(attrs={'class': 'form-control', 'id': 'password'}), \
                               min_length=8, label='Пароль:')
    password2 = forms.CharField(widget=forms.PasswordInput(attrs={'class': 'form-control', 'id': 'password2'}), \
                                min_length=8, label='Повторите пароль:')

    def clean_password2(self):
        if self.cleaned_data.get('password') != self.cleaned_data.get('password2'):
            raise forms.ValidationError('Passwords does not match')

    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_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
```

### Файл views.py:

```
from django.shortcuts import render, redirect
from my_lab7_1.forms import *
from django.contrib.auth import authenticate, logout
from django.contrib import auth
from django.contrib.auth.decorators import login_required

def signin(request):
```

```

if request.method == 'POST':
    form = RegistrationForm(request.POST)
    if form.is_valid():
        form.save()
        return redirect('success/')
    return render(request, 'signin.html', {'form': form})
else:
    form = RegistrationForm()
    return render(request, 'signin.html', {'form': form})

def signin_success(request):
    return render(request, 'signin_success.html')

def signup(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, 'signup.html', {'errors': errors,
'username':username})
    return render(request, 'signup.html', {'errors': errors})

@login_required
def signup_success(request):
    return render(request, 'signup_success.html')

@login_required
def logout_view(request):
    logout(request)
    return render(request, 'logout.html')

```

### Файл url.py:

```

from django.conf.urls import url
from django.contrib import admin
from my_lab7_1.views import *

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^signin/$', signin, name='signin'),
    url(r'^signin/success/$', signin_success),
    url(r'^signup/$', signup, name='signup'),
    url(r'^signup/success/$', signup_success),
    url(r'^logout/$', logout_view, name='logout')
]

```

### Изменения в файле settings.py:

```

LOGIN_URL = 'signup'

AUTH_USER_MODEL = 'my_lab7_1.User'

```

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

### Файл base.html:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel='stylesheet' type = 'text/css' href='{% static
"my_lab5/css/bootstrap.css" %}'>
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
<div class="jumbotron">
    <div class="container">
        {% block title_visible %}{% endblock %}
    </div>
</div>
<div class="container-fluid">
    {% block body %}{% endblock %}
</div>
</body>
</html>
```

### Файл logout.html:

```
{% extends 'base.html' %}

{% block title %}Logout{% endblock %}

{% block title_visible %}<h1>Выход</h1>{% endblock %}

{% block body %}
    <p>До свидания!</p>
{% endblock %}
```

### Файл signin.html:

```
{% extends 'base.html' %}

{% block title %}Signin{% endblock %}

{% block title_visible %}<h1>Регистрация</h1>{% endblock %}

{% block body %}
<form class="form-horizontal" action="" method="POST">
{% csrf_token %}
    <div class="form-group">
        {% if form.errors %}
            <div class="alert alert-danger">
                {% for f in form %}
                    {% if f.errors %}
                        {% for e in f.errors %}
                            <p align="center"><strong>{{ f.label }} {{ e
}}</strong></p>
            </div>
        {% endfor %}
    </div>
</form>
```

```

        {% endfor %}
    {% endif %}
{% endfor %}
</div>
{% endif %}
</div>
<div class="form-group">
    <label class="col-xs-2 control-label">Логин:</label>
    <div class="col-xs-3 input-group-sm">
        {{ form.username }}
    </div>
</div>
<div class="form-group">
    <label class="col-xs-2 control-label">Имя:</label>
    <div class="col-xs-3 input-group-sm">
        {{ form.name }}
    </div>
</div>
<div class="form-group">
    <label for="enterSurname" class="col-xs-2 control-label">Фамилия:</label>
    <div class="col-xs-3 input-group-sm">
        {{ form.surname }}
    </div>
</div>
<div class="form-group">
    <label class="col-xs-2 control-label">Email:</label>
    <div class="col-xs-3 input-group-sm">
        {{ form.email }}
    </div>
</div>
<div class="form-group">
    <label for="inputPassword" class="col-xs-2 control-label">Пароль:</label>
    <div class="col-xs-3 input-group-sm">
        {{ form.password }}
    </div>
</div>
<div class="form-group">
    <label for="confirmPassword" class="col-xs-2 control-label">Повторите
пароль:</label>
    <div class="col-xs-3 input-group-sm">
        {{ form.password2 }}
    </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 'signup' %}" class="btn btn-default" style="float:
right">Войти</a>
    </div>
</div>
</form >
{% endblock %}

```

### Файл signup.html:

```

{% extends 'base.html' %}

{% block title %}Signup{% endblock %}

{% block title_visible %}<h1>Авторизация</h1>{% endblock %}

{% block body %}

```

```

<form class="form-horizontal" action="" method="POST">
{% csrf_token %}
  <div class="form-group">
    {% for el in errors %}
      <div class="alert alert-danger" role="alert">
        <strong>{{ el }}</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 'signin' %}" class="btn btn-default" style="float:
right">Зарегистрироваться</a>
    </div>
  </div>
</form >
{% endblock %}

```

### Файл signup\_success.html:

```

{% extends 'base.html' %}

{% block title %}Success{% endblock %}

{% block title_visible %}<h1>Завершение авторизации</h1>{% endblock %}

{% block body %}
  <p>Вы авторизованы!</p>
  <a href="{% url 'logout' %}" class="btn btn-primary">Выйти</a>
{% endblock %}

```

### Файл signin\_success.html:

```

{% extends 'base.html' %}

{% block title %}Success{% endblock %}

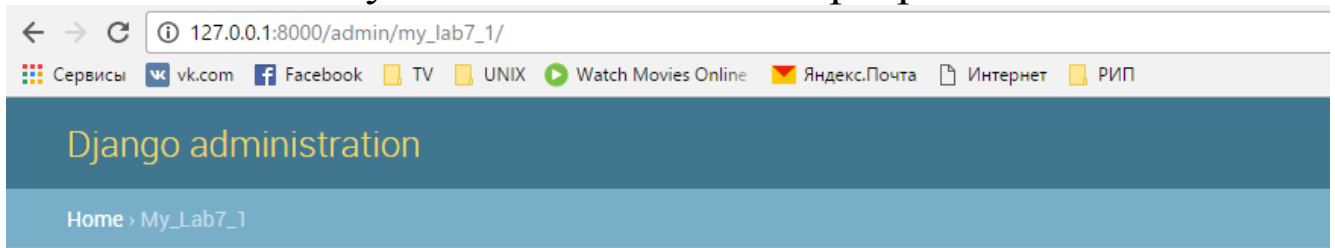
{% block title_visible %}<h1>Завершение регистрации</h1>{% endblock %}

{% block body %}
  <p>Вы зарегистрированы!</p>
{% endblock %}

```

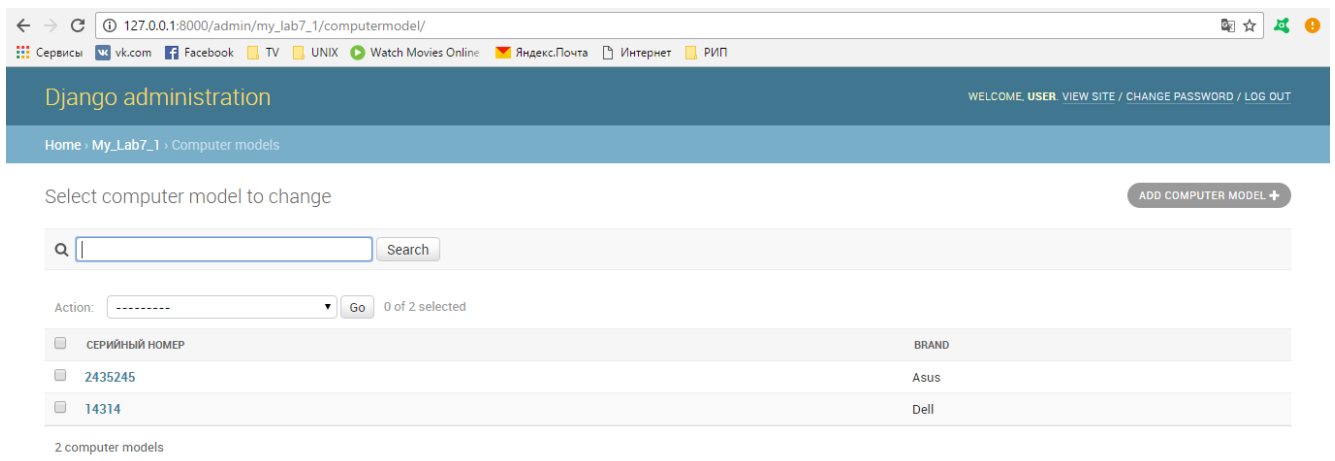


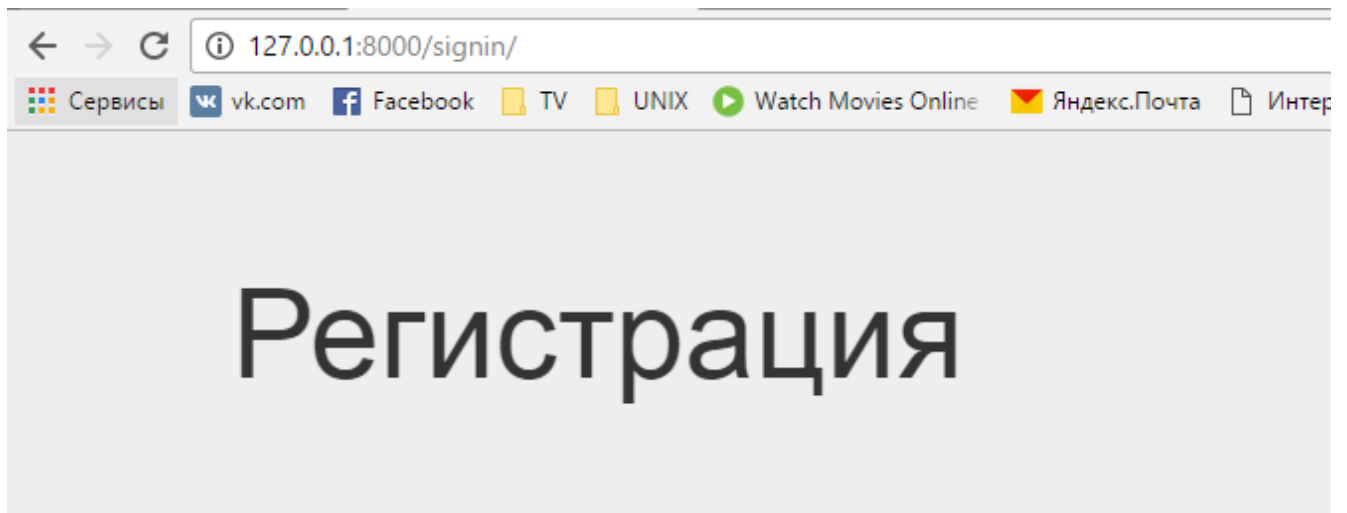
# Результаты выполнения программы



## My\_Lab7\_1 administration

MY_LAB7_1		
Computer models	<a href="#">+ Add</a>	<a href="#">Change</a>
Order models	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>





Логин:

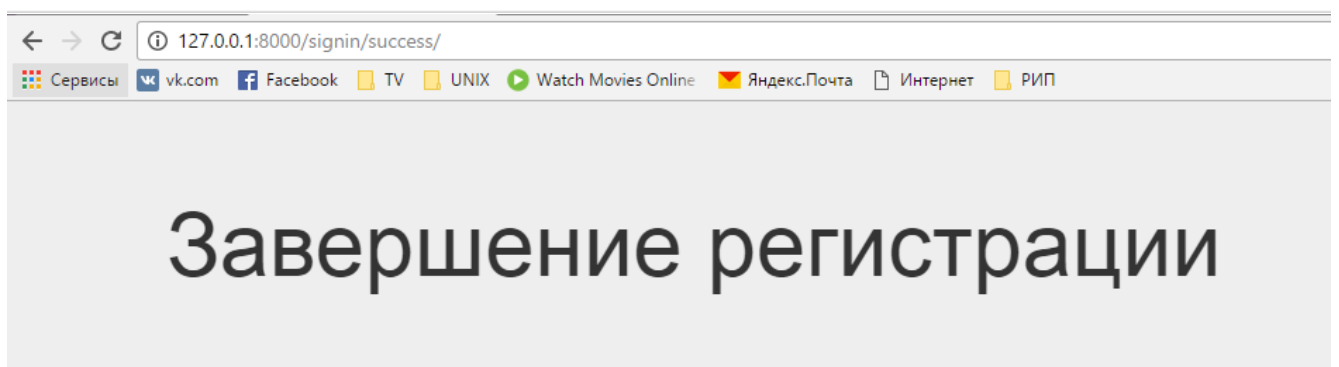
Имя:

Фамилия:

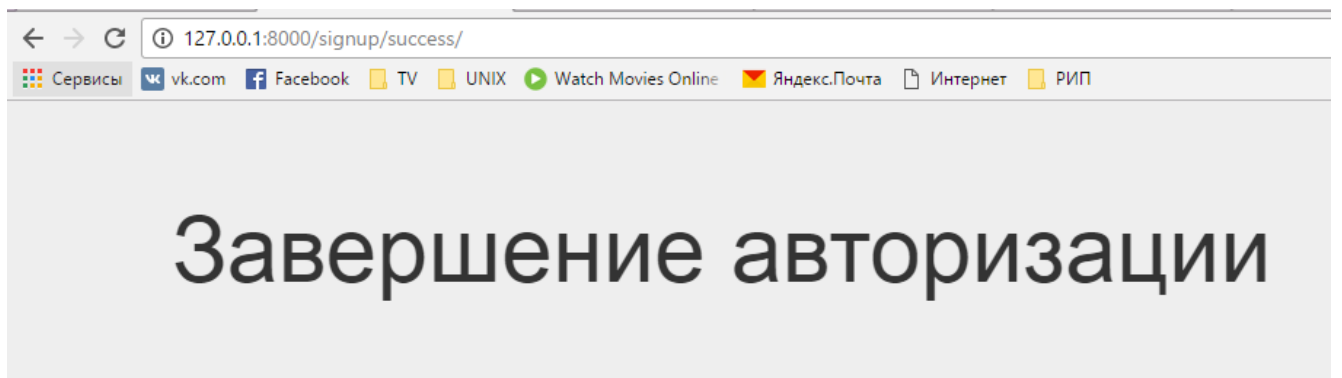
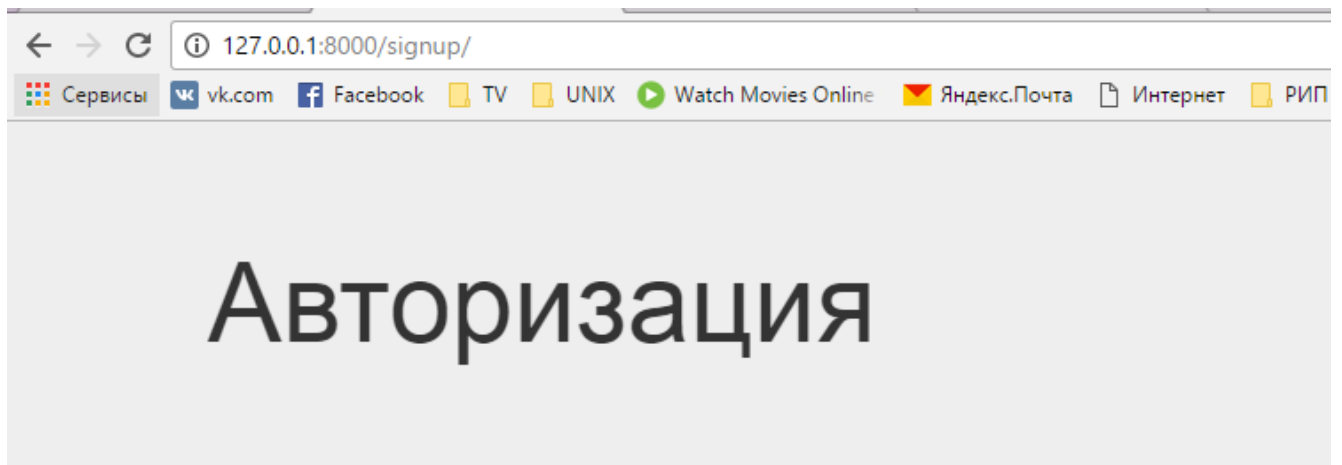
Email:

Пароль:

Повторите пароль:

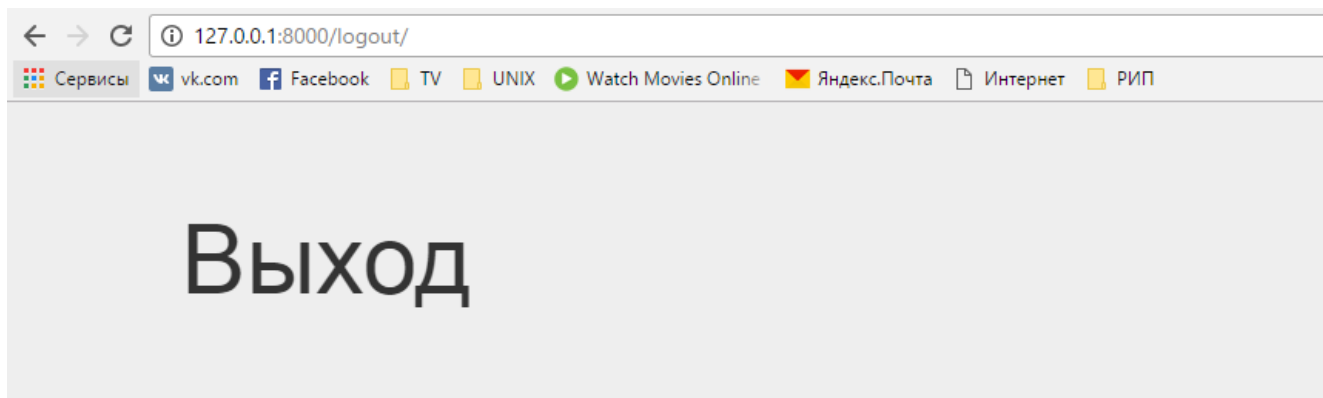


Вы зарегистрированы!



Вы авторизованы!

[Выйти](#)



До свидания!