

**Московский Государственный Технический Университет им. Н.Э.
Баумана**

Разработка интернет-приложений

Отчёт по лабораторной работе №7

«Работа с формами, авторизация»

Выполнил:

студент группы ИУ5-54

Черната Н.С.

1. Цель работы

Реализовать авторизацию с ручной валидацией (вариант 1) и с помощью Django Forms (вариант 2).

2. Листинг

Модуль views.py – ручная валидация данных POST запроса (только регистрация)

```
# coding=utf-8

from django.shortcuts import render
from django.shortcuts import render_to_response
from django.template.loader import render_to_string
from django.http import HttpResponseRedirect
from django.http import HttpResponseRedirect
from django.template import RequestContext, loader
from .models import Article
from django.contrib.auth.models import User
# Create your views here..
def index(request):
    articles = Article.objects.all()
    content = {
        'articles': articles
    }
    for a in articles:
        print(a)
    return render(request, 'articles.html', content)
def register(request):
    errors = []
    formdata = {}
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors.append("Введите имя пользователя")
        elif len(username) < 5:
            errors.append("Имя пользователя должно содержать не менее 5 символов")
        email = request.POST.get('email')
        if not email:
            errors.append("Введите адрес эл. почты")
        firstname = request.POST.get('firstname')
        if not firstname:
            errors.append("Введите своё имя")
        else:
            formdata['firstname'] = firstname
        lastname = request.POST.get('lastname')
        if not lastname:
            errors.append("Введите своё фамилию")
        else:
            formdata['lastname'] = lastname
        password = request.POST.get('password')
        if not password:
```

```

        errors.append("Введите пароль")
    elif len(password) < 8:
        errors.append("Пароль должен содержать не менее 8 символов")
    else:
        confirmpass = request.POST.get('confirmpass')
        if not confirmpass:
            errors.append("Подтвердите пароль")
        elif password != confirmpass:
            errors.append("Пароли не совпадают")
            formdata['confirmpass'] = confirmpass
        formdata['password'] = password
    sameusers = []
    try:
        sameusers.append(User.objects.get(username=username))
    except User.DoesNotExist:
        formdata['username'] = username
    try:
        sameusers.append(User.objects.get(email=email))
    except User.DoesNotExist:
        formdata['email'] = email
    if sameusers:
        errors.append("Пользователь с таким именем или адресом эл.
почты уже существует")
    if errors:
        return render(request, 'register.html', {'errors': errors,
'formdata': formdata})
    User.objects.create_user(username=username, email=email,
password=password)
    return HttpResponseRedirect("/login/")
    return render(request, 'register.html', {'errors': [], 'formdata':
formdata})
def login(request):
    return render(request, 'login.html')
def article(request, id):
    art = Article.objects.get(id=int(id))
    content = {
        'article' : art
    }
    return render(request, 'single.html', content)

```

Модуль views.py – использование Django Forms (регистрация + логин).

coding=utf-8

```

from django.shortcuts import render
from django.shortcuts import render_to_response
from django.template.loader import render_to_string
from django.http import HttpResponseRedirect
from django.http import HttpResponseRedirect
from django.template import RequestContext, loader
from .models import Article
from django.contrib.auth.models import User

```

```

from django.contrib.auth import authenticate, login as l_in, logout as l_out
import forms
# Create your views here..

def _get_user_context(request):
    context = {}
    if request.user.is_authenticated():
        context['user_logged_in'] = True
        context['username'] = request.user.username
    else:
        context['user_logged_in'] = False
    return context

def index(request):
    context = _get_user_context(request)
    articles = Article.objects.all()
    context['articles'] = articles
    return render(request, 'articles.html', context)

def register(request):
    context = _get_user_context(request)
    errors = []
    if context['user_logged_in']:
        return HttpResponseRedirect('/')
    if request.method == 'POST':
        form = forms.RegistrationForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            email = form.cleaned_data['email']
            firstname = form.cleaned_data['firstname']
            lastname = form.cleaned_data['lastname']
            password = form.cleaned_data['password']
            confirmpass = form.cleaned_data['confirmpass']
            if password != confirmpass:
                errors.append("Пароли не совпадают")
            data = form.data.copy()
            same = None
            try:
                same = User.objects.get(username=username)
            except User.DoesNotExist:
                pass
            if same:
                errors.append("Пользователь с таким именем уже существует")
                data['username'] = ""
            same = None
            try:
                same = User.objects.get(email=email)
            except User.DoesNotExist:
                pass
            if same:
                errors.append("Пользователь с таким адресом эл. почты уже существует")
                data['email'] = ""
            form.data = data

```

```

    if errors:
        return render(request, 'register.html', {'form': form,
                                                    'errors': errors})
    User.objects.create_user(username=username, email=email,
password=password, first_name=firstname, last_name=lastname)
    return HttpResponseRedirect("/login/")
    return render(request, 'register.html', {'form':
forms.RegistrationForm(), 'errors': []})
def login(request):
    context = _get_user_context(request)
    errors = []
    if context["user_logged_in"]:
        return HttpResponseRedirect('/')
    if request.method == 'POST':
        form = forms.LoginForm(request.POST)
        if form.is_valid():
            username_or_email = form.cleaned_data['username_or_email']
            password = form.cleaned_data['password']
            user = None
            try:
                user = User.objects.get(username=username_or_email)
            except User.DoesNotExist:
                try:
                    user = User.objects.get(email=username_or_email)
                except User.DoesNotExist:
                    errors.append("Пользователя с таким именем или адресом эл. почты
не существует")
            return render(request, 'login.html', {'form': form, 'errors':
errors})
            user_auth = authenticate(username=user.username, password=password)
            if user_auth is not None:
                l_in(request, user_auth)
                return HttpResponseRedirect("/success/")
            else:
                errors.append("Пароль неверен")
            return render(request, 'login.html', {'form': form, 'errors':
errors})
        return render(request, 'login.html', {'form': forms.LoginForm()})
def success(request):
    context = _get_user_context(request)
    if request.user.is_authenticated():
        context['success'] = True
        return render(request, 'success.html', context)
    else:
        return render(request, 'success.html', {"success": False})
def logout(request):
    if request.user.is_authenticated():
        l_out(request)
        return HttpResponseRedirect('/login/')
    else:
        return HttpResponseRedirect('/')
def article(request, id):
    context = _get_user_context(request)

```

```
art = Article.objects.get(id=int(id))
context['article'] = art
return render(request, 'single.html', context)
```

Шаблон register.html

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

<div class="article-body">
    {% if errors %}
        {% for error in errors %}
            <div class="error">
                {{error}}
            </div>
        {% endfor %}
    {% endif %}
    <form method="POST">
        {% csrf_token %}
        <div class="content">
            <table align="center">
                {{form.as_table}}
            </table>
            <button action="submit" class="btn btn-
primary">Зарегистрироваться</button>
        </div>
    </form>
</div>

{% endblock %}
```

Шаблон login.html аналогичен.

3. Результат

Форма регистрации:

[Main](#) [Departments](#) [Login](#) [Logout](#)

Sing up

Login

Password

Confirm your password

First name

Last name

E-mail

[Sing up](#)

Форма входа:

[Main](#) [Departments](#) [Login](#) [Logout](#)

Login

Username:

Password:

[Sing Up](#) [Login](#)

Главная страница после входа пользователя:

