**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

Геништа Михаил

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** HttpResponse

**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** HttpResponse

**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**>

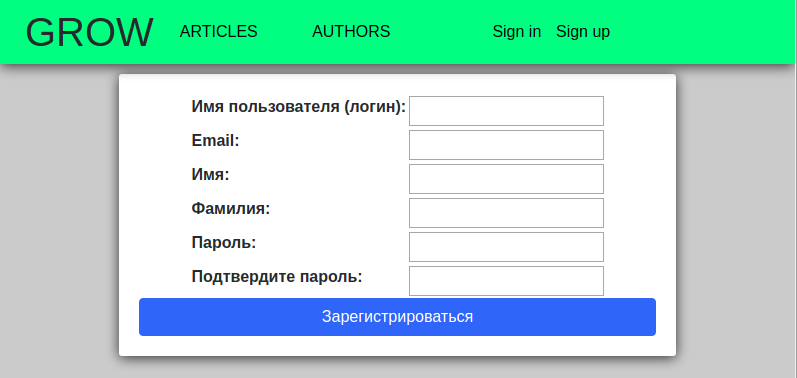
</**div**>

{% endblock %}

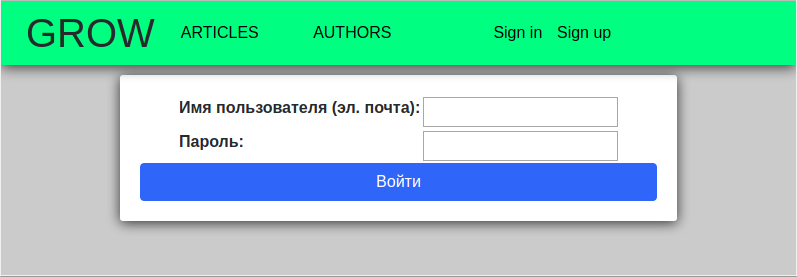
Шаблон login.html аналогичен.

3. Результат

Форма регистрации:



Форма входа:



Главная страница после входа пользователя:

