Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»

Лабораторная работа №8

по дисциплине

«Кроссплатформенные технологии программирования»

Выполнил студент

группы БСТ1702

Нурмамедов Р.Ш.

Вариант 12

Москва - 2019

**Задание №1:**

* Реализовать программу регистрации новых пользователей
* Организовать для неё веб-страницу
* Обеспечить возможность перехода к регистрации с общей страницы статей и со страниц отдельных статей

Код функции-обработчика:

def reg (request):

if not request.user.is\_anonymous():

if request.method == "POST":

# создать пользователя, если метод POST

new\_user = {

'login': request.POST["login"],

'email': request.POST["email"],

'password':request.POST["password"]

}

# в словаре new\_user будет храниться информация, введенная пользователем

if new\_user["login"] and new\_user["password"]:

# если поля заполнены без ошибок

try:

# если введённый логин уже существует, то пользователя перенаправят на страницу авторизации

er\_user=User.objects.get(username=new\_user["login"])

new\_user['errors'] = u"Пользователь с таким логином уже существует".encode('1251')

return redirect('auth')

except User.DoesNotExist:

# создаёт нового пользователя и перенаправляет его на общую страницу статей

user=User.objects.create\_user(new\_user["login"], new\_user["email"], new\_user["password"])

return redirect('archive')

else:

# если введенные данные некорректны

new\_user['errors'] = u"Не все обязатаельные поля заполнены".encode('1251')

return render(request, 'reg\_new.html', {'form': new\_user})

else:

# просто вернуть страницу с формой, если метод GET

return render(request, 'reg\_new.html', {})

else:

raise Http404

Код html-файла регистрации:

<!DOCTYPE html>

<html>

<head>

<title>Регистрация</title>

<link rel="stylesheet" href="{{STATIC\_URL}}css/article.css">

</head>

<body>

<div class="content">

<h1>Введите данные для регистрации</h1>

<form method="POST">{% csrf\_token %}

<h2>Введите логин (обязательное поле !!!):</h2>

<div class="new-login">

<input type="text" name="login" placeholder="Login" value="{{ form.title }}">

</div>

<h2>Введите адрес электронной почты:</h2>

<div class="new-email">

<input type="text" name="email" placeholder="E-mail" value="{{ form.title }}">

</div>

<h2>Введите пароль (обязательное поле !!!):</h2>

<div class="new-password">

<input type="text" name="password" placeholder="Password" value="{{ form.title }}">

</div>

<div class="form-button">

<input type="submit" value="Сохранить">

</div>

</form>

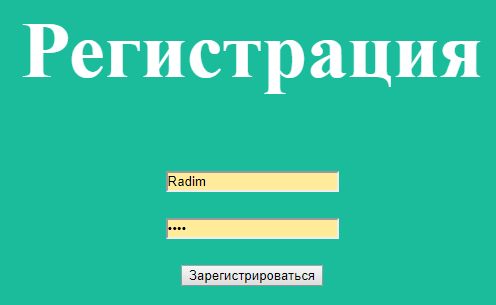
{{ form.errors }}

</div>

</body>

</html>

Результат:



При введении уже имеющегося логина пользователь будет перенаправлен на страницу авторизации. При введении некорректных данных страница просто обновится. При корректном введении новых данных пользователь будет перенаправлен на общую страницу статей.

**Задание №2:**

* Реализовать программу авторизации пользователя
* Организовать для неё веб-страницу
* Обеспечить проверку введённых данных на наличие в системе

Код функции-обработчика:

def auth (request):

if not request.user.is\_anonymous():

if request.method == "POST":

# авторизация пользователя, если метод POST

auth\_user = {

'login': request.POST["login"],

'password':request.POST["password"]

}

# в словаре auth\_user будет храниться информация, введенная пользователем

if auth\_user["login"] and auth\_user["password"]:

try:

Log=User.objects.get(username=auth\_user["login"])

if check\_password(auth\_user["password"],Log.password):

# если поля заполнены без ошибок то пользователь будет авторизован и перенаправлен на общую страницу статей

user = authenticate(username=auth\_user["login"], password=auth\_user["password"])

login(request, user)

return redirect('archive')

else:

auth\_user['errors'] = u"Вы ввели неверный пароль".encode('1251')

return render(request,'auth.html',{'form':auth\_user})

except User.DoesNotExist:

auth\_user['errors'] = u"Вы ввели неверный логин".encode('1251')

return render(request,'auth.html',{'form':auth\_user})

else:

# если введенные данные некорректны

auth\_user['errors'] = u"Не все обязатаельные поля заполнены".encode('1251')

return render(request, 'auth.html', {'form': auth})

else:

# просто вернуть страницу с формой, если метод GET

return render(request, 'auth.html', {})

else:

raise Http404

Код html-файла авторизации:

<!DOCTYPE html>

<html>

<head>

<title>Авторизация</title>

<link rel="stylesheet" href="{{STATIC\_URL}}css/article.css">

</head>

<body>

<div class="content">

<h1>Авторизация</h1>

<form method="POST">{% csrf\_token %}

<h2>Введите логин:</h2>

<div class="auth-login">

<input type="text" name="login" placeholder="Login" value="{{ form.title }}">

</div>

<h2>Введите пароль:</h2>

<div class="auth-password">

<input type="text" name="password" placeholder="Password" value="{{ form.title }}">

</div>

<div class="form-button">

<input type="submit" value="Сохранить">

</div>

</form>

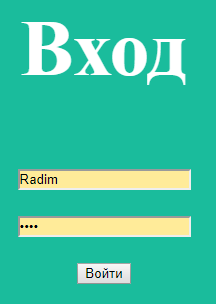
{{ form.errors }}

</div>

</body>

</html>

Результат:



При введении неверного логина или пароля страница просто обновится, иначе пользователь будет перенаправлен на общую страницу статей.