

**Ministerul Educației, Culturii și Cercetării al Republicii Moldova**  
**Universitatea Tehnică a Moldovei**  
**Facultatea Calculatoare Informatică și Microelectronică**  
**Departamentul Ingineria Software și Automatică**

Disciplina: Tehnologii Web

**Lucrarea de laborator nr.3**  
**Tema: Model de proiectare. Pattern BusinessLogic**

A efectuat studentul: Eric Capațina gr. TI-184  
A verificat: asist.univ Cristian Rusu

## Chișinău 2020

Scopul lucrării:

Implementarea BusinessLogic în proiectul creat.

Crearea sistemului de înregistrare și legăturii NewUser – Database.

Pentru a crea sistemul de înregistrare al utilizatorului nou, creăm în proiect fișierul "database.py", ce va conține următoarea secvență de cod:

```
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm
from . import models

class UserRegistrationForm(UserCreationForm):
    email = forms.EmailField()

    class Meta:
        model = User
        fields = ['username', 'password1', 'password2', 'email']
        help_texts = {
            'username': None,
            'email': None,
        }
}
```

În cazul dat, utilizatorul va fi nevoit să completeze următoarele: Username, parola, parola repetată și E-mailu. Pentru a scoate textul ajutător implementat de framework, folosim "help\_texts" și la field-ul ales îi dăm valoarea "None"

În "views.py" schimbăm funcția creată anterior pentru accesarea documentului HTML de înregistrare:

```
from .database import UserRegistrationForm
```

Implementăm clasa din "database.py"

```
def registration(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            messages.success(request, f'Oh, HI {username}!')
            return redirect(home)
    else:
        form = UserRegistrationForm()
    return render(request, 'lab1reg.html', {'form': form})
```

Specificăm metoda ce va fi folosită în documentul HTML, egalăm "form" cu clasa implementată din "database.py" pentru ca field-urile din form să fie anume acelea, care sunt scrise în clasă.

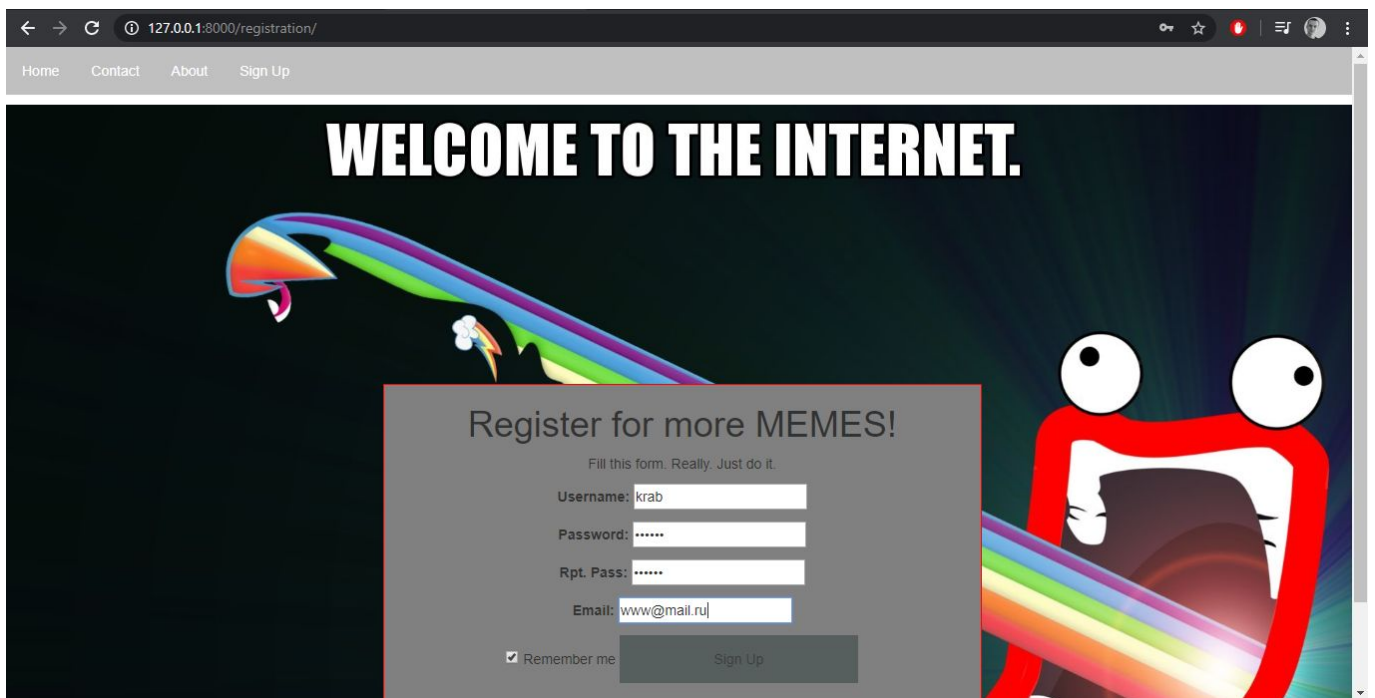


Figura 1 Panela de înregistrare

Introducem datele și verificăm baza de date a serverului, dacă useru a fost creat.

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	krab	www@mail.ru			✓
<input type="checkbox"/>	test				✗
<input type="checkbox"/>	test123				✗
3 users					

Figura 2 Baza de date cu utilizatorii înregistrați

Crearea Log-in și Log-out:

Adaugăm următoarele secvențe de cod în "urls.py":

```
from django.contrib.auth import views as auth_views
```

Implementăm biblioteca

```
urlpatterns = [
    path('registration/', views.registration, name='registration'),
    path('signup/', auth_views.LoginView.as_view(template_name='singup.html'),
name='login'),
    path('logout/', auth_views.LogoutView.as_view(template_name='home.html'),
name='logout'),
    path('', views.home, name='home')
]
```

Adaugăm funcții noi pentru accesarea documentelor HTML în "views.py":

```
def signup(request):
    return render(request, 'singup.html')
def home(request):
    return render(request, 'home.html')
```

Creăm documente HTML și CSS pentru pagina LogIn:

```
{% load static %}
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{% static 'css/login.css' %}">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
    <title>Pepe LogIn</title>
    <style>
</style>
</head>
<body>
<ul>
    <li><a class="active" href="/home">Home</a></li>
    <li><a href="/registration">Registration</a></li>
    <li><a href="/contact">Contact</a></li>
    <li><a href="/home">About</a></li>
</ul>
<div class="background">
    <form method="POST" class="div2">
        <h1>Welcome back, MemeLord.</h1>
        {% csrf_token %}
        {{ form.as_p }}
        <div class="form-group">
            <input type="checkbox" checked="checked" name="remember"
style="margin-bottom:15px"> Remember me</label>
            <button class="signupbtn" type="submit">LogIn</button>
        </div>
    </form>
</div>
</div>
</body>
<script>
function myFunction()
{
    document.getElementById("id01").style.background = "purple";
    location.href = "signup.html";
}
</script>
</html>
```

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
```

```
    background-color: silver;
}
button {
    background-color: #515A5A;
    color: black;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 40%;
    opacity: 0.9;
}
li {
    float: left;
}

li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

li a:hover {
    background-color: gray;
}

.div2 {
    background-color: gray;
    width: 600px;
    height: 340px;
    position: absolute;
    border: 1px solid red;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
    left: 28%;
    top: 40%;
}

p.date {
    text-align: center;
}

.background {
    background-image: url('https://i.imgur.com/0BTnEB2.jpg');
    background-size: cover;
    height: 700px;
    position: relative;
}
```

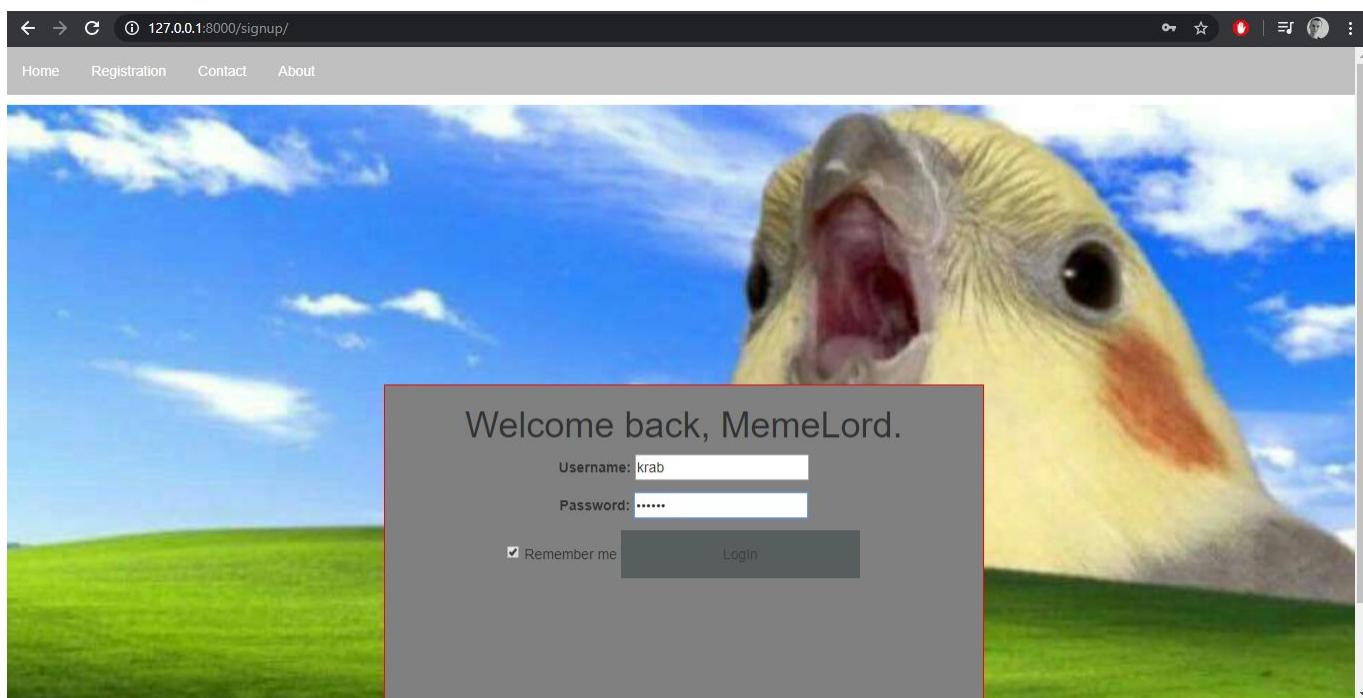


Figura 3 Pagina LogIn



Figura 4 Panela utilizatorului ce nu e logat



Figura 5 Panela utilizatorului logat

### Concluzia:

În lucrarea de laborator nr.3 am făcut cunoștința cu crearea legăturilor cu baza de date în framework. Am implementat sistema de înregistrare, login și logout în proiectul nostru.

### GitHub:

<https://github.com/EricCapatina/TehWebLaborator>