

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.4
Tema: Crearea logicii. Integrarea bazei de date.

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

Chișinău 2020

Scopul lucrării:

Introducerea a bazei de date, crearea postărilor.

Pentru a crea postările și a le lega de baza de date avem nevoie să creăm fișier nou în proiect – ”modules.py” și adăugăm următoarea secvență de cod:

```
from django.db import models
from django.contrib.auth.models import User
from ckeditor.fields import RichTextField
class Notes(models.Model):
    title = models.CharField(null=False, max_length=100)
    body = RichTextField(blank=True)
    date = models.DateTimeField(auto_now_add=True)
    author = models.ForeignKey(User, null=False, default=None, on_delete=models.CASCADE)

    def __str__(self):
        return self.title
```

În ”database.py” implementăm ”modules.py” și clasa Notes. Adăugăm clasa pentru crearea field-urilor postărilor:

```
class CreateNote(forms.ModelForm):
    class Meta:
        model = models.Notes
        fields = ['title', 'body']
```

În ”views.py” creăm funcție pentru a accesa clasa Notes și documentele HTML necesare:

```
def notes_detail(request):
    note = Notes.objects.all().order_by('date');
    return render(request, 'create_note.html', { 'note': note })

def note_create(request):
    if request.method == 'POST':
        form = forms.CreateNote(request.POST, request.FILES)
        if form.is_valid():
            instance = form.save(commit=False)
            instance.author = request.user
            instance.save()
            return render(request, 'home.html')
    else:
        form = forms.CreateNote()
    return render(request, 'create.html', { 'form': form })
```

Creăm documentele HTML:

Pentru a crea postare nouă:

```
<head>
    <meta charset="UTF-8">
    <title>Create </title>
</head>
```

```

<body>
<a href="/home">Home</a>
<a href="/create">View Posts</a>
<div class="create-article">
  <h2>Create Meme Post</h2>
  <form class="site-form" method="POST">
    {%csrf_token%}
    {{form.media}}
    {{form.as_p}}
    <input type="submit" value="Create Note">
  </form>
</div>
</body>
</html>

```

Pentru a verifica postările ce există:

```

<head>

</head>
<body>
<h1>MemeLord Posts</h1>
<a href="/home">Home</a>
<a href="/create_note">Create another</a>
  <div class="note">
    {% for notes in note %}
      <div class="notes">
        <h2>{{ notes.title }}</h2>
        <p>{{ notes.body|safe }}</p>
        <p class="author">added by MemeLord {{ notes.author.username }}</p>
        <p>Date'n'Time: {{ notes.date }}</p>
      </div>
    {% endfor %}
  </div>
</body>
</html>

```

În "urls.py" adăugăm următoarele secvențe pentru a crea path-ul spre documentele HTML:

```

path('create/', views.notes_detail, name="create"),
path('create_note/', views.note_create, name='create_note'),

```

În terminal aplicăm la "python manage.py makemigrations" pentru a crea clasa "Notes" în baza de date:

```

(venv) D:\Pycharm\TehWeb\RADIPACANOV>python manage.py makemigrations
Migrations for 'RADIPACANOV':
  RADIPACANOV\migrations\0001_initial.py
  - Create model Notes

```

Figura 1 Crearea "Notes" în baza de date

Codul în baza de date după clasa creată:

```

operations = [
    migrations.CreateModel(
        name='Notes',
        fields=[
            ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
verbose_name='ID')),
            ('title', models.CharField(max_length=100)),
            ('body', ckeditor.fields.RichTextField(blank=True)),
            ('date', models.DateTimeField(auto_now_add=True)),
            ('author', models.ForeignKey(default=None,
on_delete=django.db.models.deletion.CASCADE, to=settings.AUTH_USER_MODEL)),
        ],
    ),
]

```

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/create_note/'. The page title is 'Create Meme Post'. Below the title, there is a form with two main sections: 'Title:' and 'Body:'. The 'Title:' field contains the text 'LaboratorTehWeb'. The 'Body:' field is a rich text editor with a toolbar containing various icons for text formatting (bold, italic, underline, strikethrough, text color, background color, link, unlink, list, indent, outdent, etc.) and a 'Create Post' button at the bottom. The text area of the rich text editor contains the text 'Verificarea creării unei postări'.

Figura 1 Pagina creării unei postări

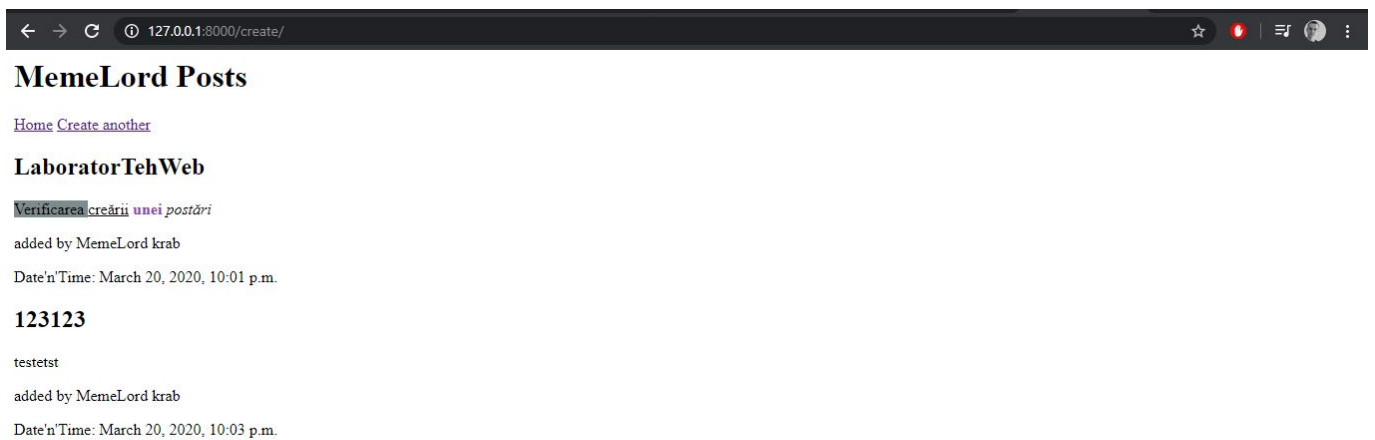


Figura 2 Pagina cu toate postările create



Figura 3 Verificăm dacă postările se salvează în baza de date

Funcția EmailSend:

Pentru a automatiza transmiterea mesajelor prin Email de la site-ul nostru spre utilizator vom face următoarele schimbări:

În setările "settings.py" adăugăm următoarele secvențe de cod:

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER = 'GMAILUMEU' # parola cu mail se schimba
EMAIL_HOST_PASSWORD = 'PAROLAMEA'
```

Acestea fiind date care introducem și pe care le va folosi serverul pentru a transmite mesajul E-mail.

Creăm path-ul nou în "urls.py":

```
path('contact/', views.contact, name='contact'),
```

Creăm document HTML:

```
{% load static %}
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="{% static 'css/contact.css' %}">
    <title>E-mail message from MemeLord-BOSS</title>
</head>
<body>
<div class="background">
    <div class="div2">
<h1>Contact a MemeLord-BOSS through ur pathetic E-Mail</h1>
<a href="/contact">Please contact me, MemeLord-BOSS!</a>
<p><a href="/home">Home</a></p>
    </div>
</div>
</body>
</html>
```

Creăm funcția de transmiterea mesajelor în "views.py":

```
from django.core.mail import send_mail
```

Implementăm biblioteca

```
def emailsend(request):
    send_mail('A message for MemeLord',
            'Stay strong, King. This is an automated message for you from your kingdom
called "Me mi".',
            'eric199k@gmail.com',
            [forms.EmailField()],
            fail_silently=False)
    return render(request, 'emailsend.html')
```

În cazul dat se va transmite mesaj de la "eric199k@gmail.com" spre mailu utilizatorului. Titlu și textul mesajului sunt scrise înainte de aceasta.

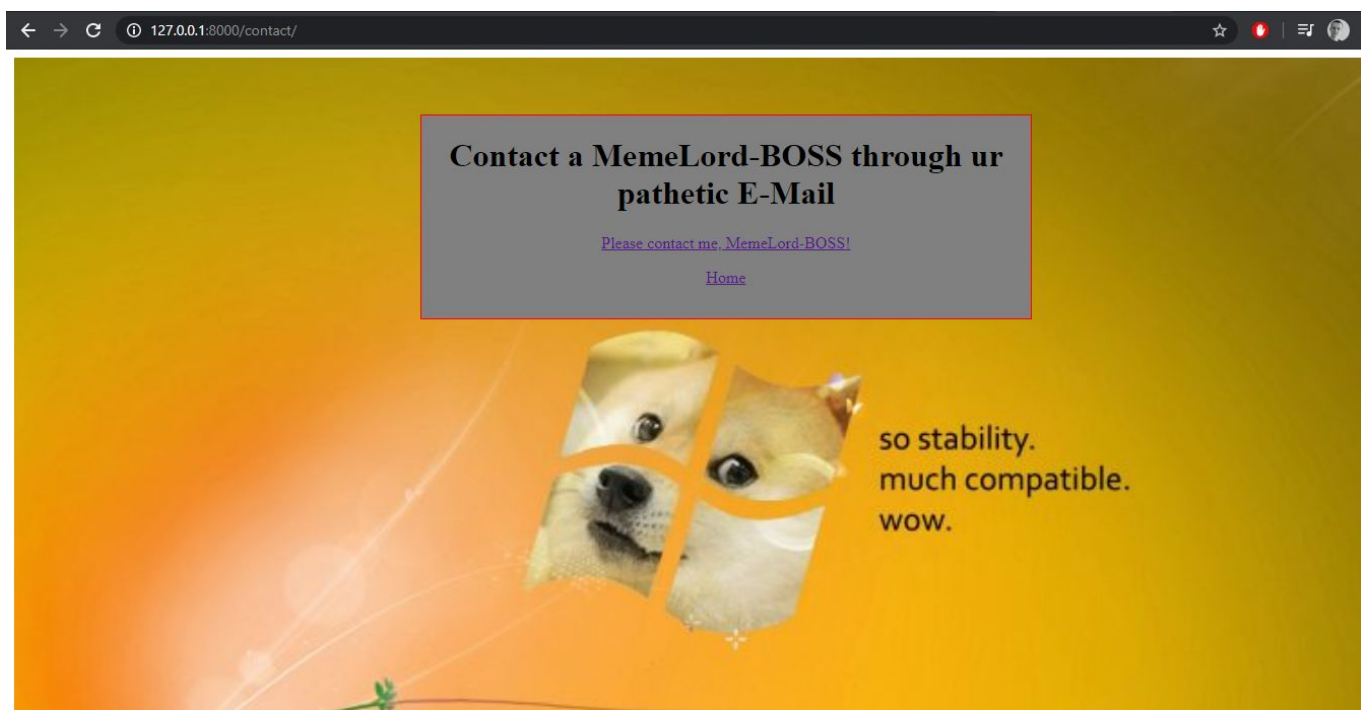


Figura 4 Pagina de contactare prin Email

În urma accesării butonului "Please contact me MemeLord-BOSS!" de pe pagina de contactare, utilizatorul, în cazul dat "wit98_99@mail.ru" va primi mesajul.

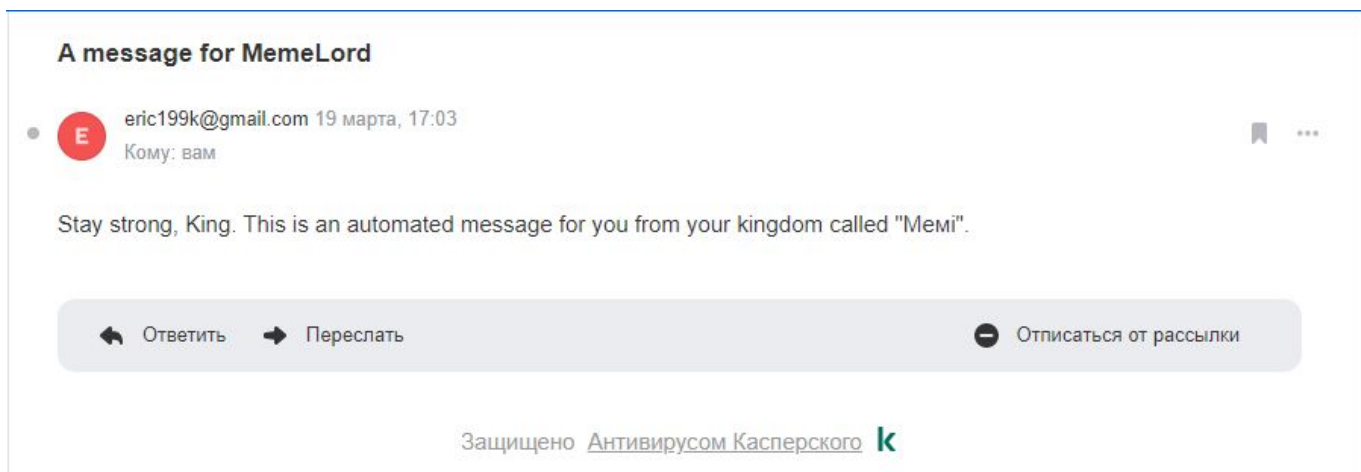


Figura 5 Mesajul primit dupa accesarea butonului de pe pagina de contactare

Concluzia:

În urma efectuării lucrării de laborator nr.4, am făcut cunoștință cu lucru în baza de date, crearea bazei de date și implementarea anumitor funcții ce necesită aceasta bază. Am creat posibilitatea ca utilizatorul să creeze postări, aceste fiind salvate în baza de date și fiind accesibile oricând. Am făcut cunoștință cu posibilitatea de a transmite mesajul automat prin E-mail de la site spre utilizator.