

Flask

Partea III. Formulare în Flask

Ce ne așteaptă?

1. Bazele formularelor în Flask
2. Câmpurile formularelor în Flask
3. Alerte Flash

1. Bazele formularelor în Flask

Esența formularelor în Flask

- Flask include unele clase pentru crearea formularelor HTML direct din codul Python.
- Pentru aceasta se utilizează pachetele flask_wtf și wtforms
- Pașii parcurși pentru a crea un formular cu scriptul Python Flask.
 - Se configurează o cheie secretă pentru securitate
 - Se creează o clasă a formularului ce moștenește clasa WTForm
 - Se creează un câmp Fields pentru fiecare componentă a formularului
 - Se setează funcția view
 - Se adaugă methods = ['GET', 'POST']
 - Se creează o instanță a clasei Form
 - Se gestionează transmisiunea formularului

Scriptul paginii HTML

- În acest folder templates se creează un document HTML numit pag1.html ce va include formularul creat în script Python:
- Documentul sablon.html ce conține șablonul moștenit se consideră cel din capitolul precedent.

```
{% extends "sablon.html" %}
{% block content %}
<p>
    {% if nume %}
    Numele introdus este {{nume}} <br>
    Actualizati-l in formularul de mai jos:
    {% else %}
    Introduceti numele dvs:
    {% endif %}
</p>
<form method="POST">
    {{ form.hidden_tag() }}
    {{ form.nume.label }} {{ form.nume() }}
    {{ form.submit() }}
</form>
{% endblock %}
```

Scriptul Python

- Se modifică conținutul fișierului app1.py care va include și o clasă de crearea a formularului

```
from flask import Flask, render_template
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField

app = Flask(__name__)
app.config['SECRET_KEY'] = 'mysecretkey'

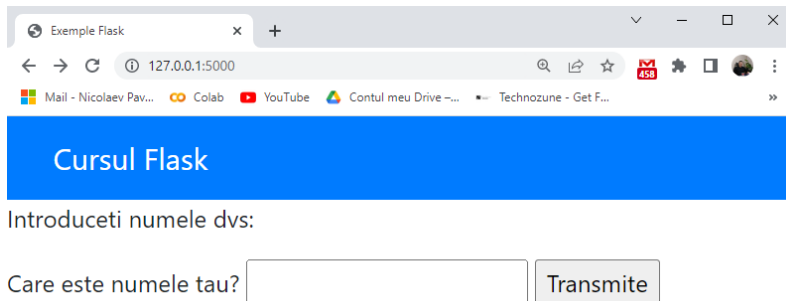
class InfoForm(FlaskForm):
    nume = StringField('Care este numele tau?')
    submit = SubmitField('Transmite')

@app.route('/', methods=['GET', 'POST'])
def pag1():
    form = InfoForm()
    if form.validate_on_submit():
        nume = form.nume.data
        form.nume.data = ''
        return render_template('pag1.html', form=form, nume=nume)

if __name__ == '__main__':
    app.run(debug=True)
```

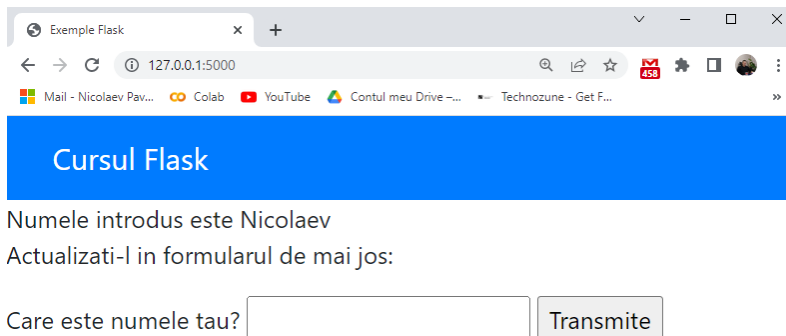
Paginile aplicație

■ Pagina de pornire până la transmiterea formularului



The screenshot shows a web browser window with the title 'Exemple Flask' and the address bar displaying '127.0.0.1:5000'. The browser's tab bar includes links to 'Mail - Nicolaev Pav...', 'Colab', 'YouTube', 'Contul meu Drive...', and 'Technozune - Get F...'. The main content area features a blue header with the text 'Cursul Flask'. Below the header, the text 'Introduceti numele dvs:' is displayed. Underneath, there is a label 'Care este numele tau?' followed by an empty text input field and a 'Transmite' button.

■ Pagina de pornire după completarea și transmiterea formularului



The screenshot shows the same web browser window as before, but with updated content. The blue header 'Cursul Flask' remains. Below it, the text 'Numele introdus este Nicolaev' is displayed, followed by 'Actualizati-l in formularul de mai jos:'. At the bottom, the label 'Care este numele tau?' is followed by the same text input field, which now contains the value 'Nicolaev', and the 'Transmite' button.

2. Câmpurile formularului în Flask

Esența câmpurilor formularului în Flask

- Fiecare câmp posibil al formularului HTML are o clasă `WTFForm` corespunzătoare ce poate fi importată
- Pachetul `wtform` include și validatori ce pot fi simplu implementați
- Validatorii permit verificarea datelor introduse în câmpurile formularului, de exemplu necesitatea completării obligatorii
- Pentru salvarea temporară a datelor extrase din formular se poate utiliza un obiect al sesiunii Flask

Scriptul Python (1)

- Se modifică conținutul fișierului app1.py care va include și o clasă de crearea a formularului cu mai multe câmpuri (va continua si pe alt slide)

```
from flask import Flask, render_template
from flask import Flask, render_template, session, redirect, url_for, session
from flask_wtf import FlaskForm
from wtforms import (StringField, BooleanField, DateTimeField, RadioField, SelectField, TextField,
                    TextAreaField, SubmitField)
from wtforms.validators import DataRequired

app = Flask(__name__)

app.config['SECRET_KEY'] = 'mysecretkey'

class InfoForm(FlaskForm):
    nume = StringField('Introduceti numele si prenumele dvs?', validators=[DataRequired()])
    python = BooleanField("Sunteti initiat in limbajul Python?")
    nivel = RadioField('Selectati nivelul dvs de cunoastere Flask:',
                     choices=[('Inceparator', 'Inceparator'), ('Intermediar', 'Intermediar'), ('Avansat', 'Avansat')])
    aplicare = SelectField(u'Selectati domeniul unde veti aplica cunostintele Flask',
                          choices=[('Portogoliu personal', 'Portogoliu personal'), ('Programarea web', 'Programarea web'),
                                    ('Data Science', 'Data Science'), ('Instruire', 'Instruire')])
    feedback = TextAreaField()
    submit = SubmitField('Transmite')
```


Scriptul Python (2)

■ Continuare de pe slide-ul precedent

```
@app.route('/', methods=['GET', 'POST'])
def pag1():
    form = InfoForm()
    if form.validate_on_submit():
        session['nume'] = form.nume.data
        session['python'] = form.python.data
        session['nivel'] = form.nivel.data
        session['aplicare'] = form.aplicare.data
        session['feedback'] = form.feedback.data
        return redirect(url_for("multumire"))
    return render_template('pag1.html', form=form)

@app.route('/multumire')
def multumire():
    return render_template('multumire.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Pagina HTML de pornire

- Se creează un șablon .html numit pag1.html pentru pagina de pornire ce va conține formularul . La transmiterea formularului datele se salvează în sesiune

```
{% extends "sablon.html" %}
{% block content %}
<h3>Bun venit la cursul Flask</h3>
<form method="POST">
  {{ form.hidden_tag() }}
  {{ form.nume.label }} {{form.nume}}
  <br>
  {{ form.python.label }} {{form.python}}
  <br>
  {{form.nivel.label}}{{form.nivel}}
  <br>
  {{form.aplicare.label}}{{form.aplicare}}
  <br>
  Transmite un feedback?
  <br>
  {{form.feedback}}
  <br>
  {{ form.submit() }}
</form>
{% endblock %}
```

Pagina HTML de mulțumire

- Se creează un șablon .html numit multumire.html care va afișa un mesaj de mulțumire și va lista datele introduse în formular și salvate în sesiune după transmiterea acestuia

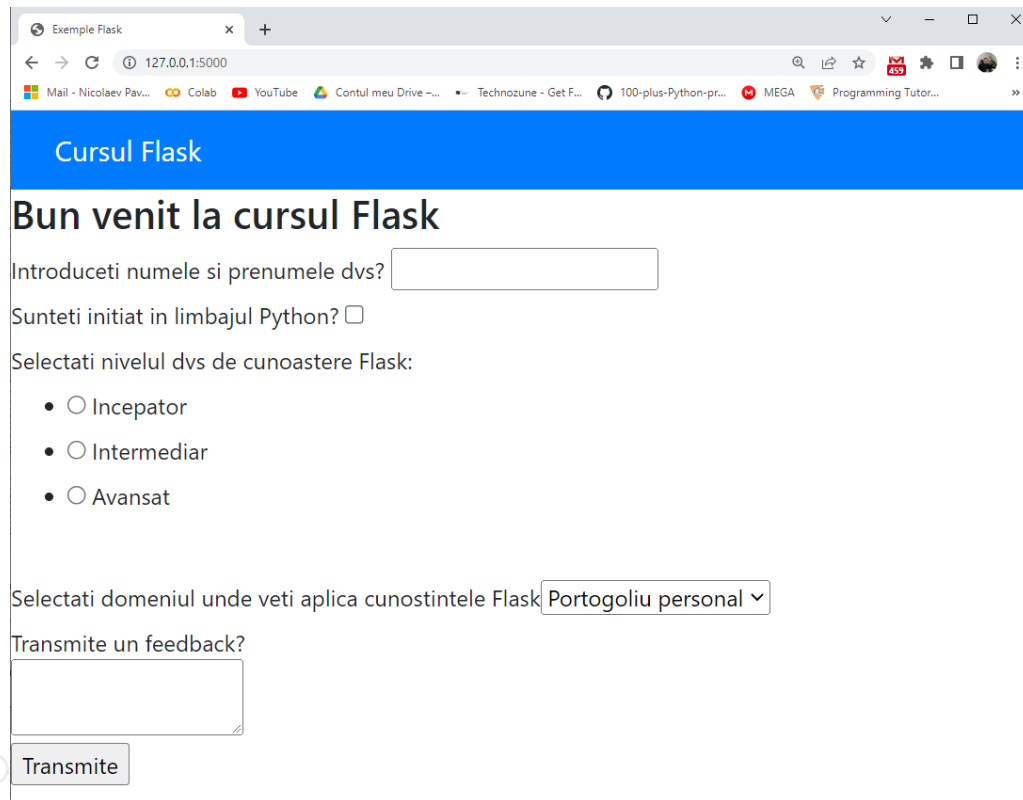
```
{% extends "sablon.html" %}

{% block content %}

<h3>Va multumim. Aici vei gasi informatia transmisa:</h3>
<ul>
  <li>Nume: {{session['nume']}}</li>
  <li>Python: {{session['python']}}</li>
  <li>Nivel: {{session['nivel']}}</li>
  <li>Aplicare: {{session['aplicare']}}</li>
  <li>Feedback: {{session['feedback']}}</li>
</ul>{% endblock %}
```

Pagina web de pornire

- În browser se introduce adresa `http://127.0.0.1:5000`



Exemple Flask

127.0.0.1:5000

Cursul Flask

Bun venit la cursul Flask

Introduceti numele si prenumele dvs?

Sunteti initiat in limbajul Python? ☐

Selectati nivelul dvs de cunoastere Flask:

- ☐ Incepator
- ☐ Intermediar
- ☐ Avansat

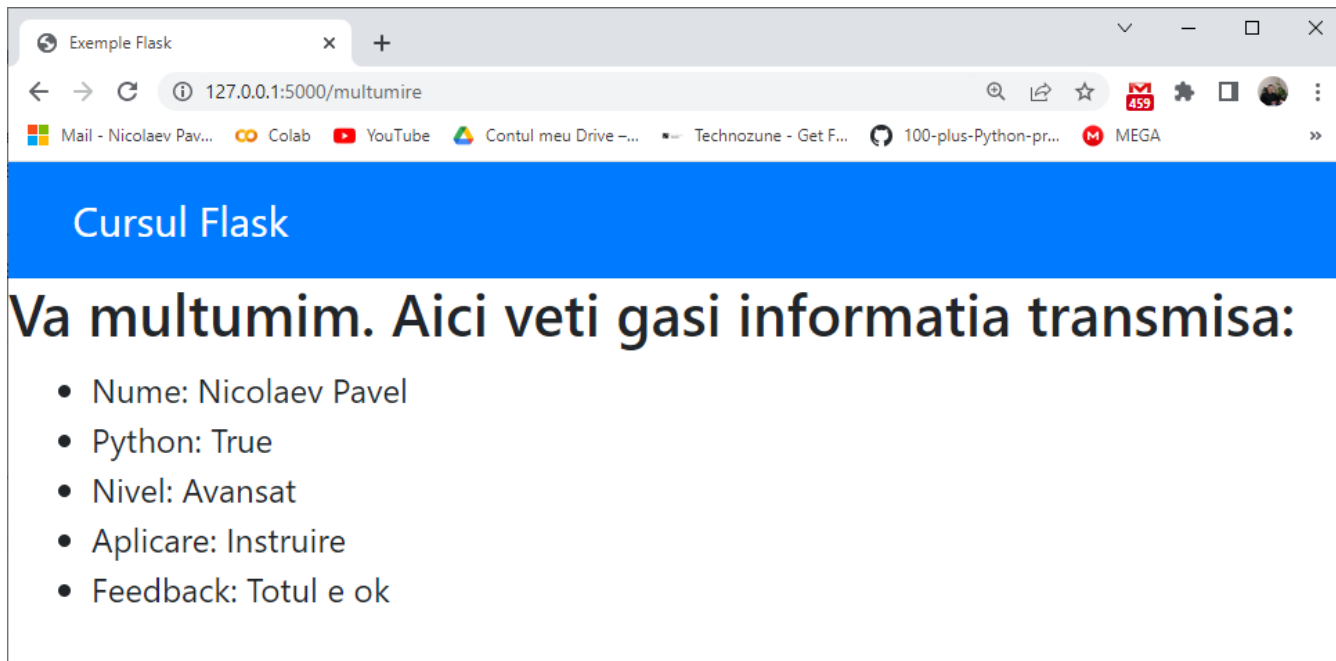
Selectati domeniul unde veti aplica cunostintele Flask

Transmite un feedback?

Transmite

Pagina web de mulțumire

- Se completează și se transmite formularul din pagina de pornire



3. Alerte Flash

Esența alertelor Flash

- Deseori, se dorește transmiterea unui un mesaj utilizatorului care nu trebuie să fie salvat sau repetat permanent pe pagina șablon.
- Se poate afișa un mesaj către utilizator care poate fi apoi închis
- Pentru acest tip de mesaje Flask utilizează funcția `flash()`
- Se creează un șablon `.html` (numit `sablon.html`) ce va conține informația utilă și altor pagini
- După apelarea acestei funcții automat se generează o lista de mesaje `get_flashed_messages()`
- Pentru stilizarea mesajului se utilizeza codul Bootstrap corespunzator alertelor (<https://getbootstrap.com/docs/4.0/components/alerts/>)

Scriptul Python

- Se modifică conținutul fișierului app1.py care va include și o clasă de crearea a unui formular și va apela o alertă transmiterea acestuia

```
from flask import Flask, render_template, flash, session, redirect, url_for
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
```

```
app = Flask(__name__)
app.config['SECRET_KEY'] = 'mysecretkey'
```

```
class SimpleForm(FlaskForm):
    nume = StringField('Introduceti prenumele dvs?')
    submit = SubmitField('Actioneza aici.')
```

```
@app.route('/', methods=['GET', 'POST'])
def pag1():
    form = SimpleForm()
    if form.validate_on_submit():
        session['nume'] = form.nume.data
        flash("Tocmai ai actionat butonul 'Actioneza aici.'")
        return redirect(url_for('pag1'))
    return render_template('pag1.html', form=form)
```

```
if __name__ == '__main__':
    app.run(debug=True)
```

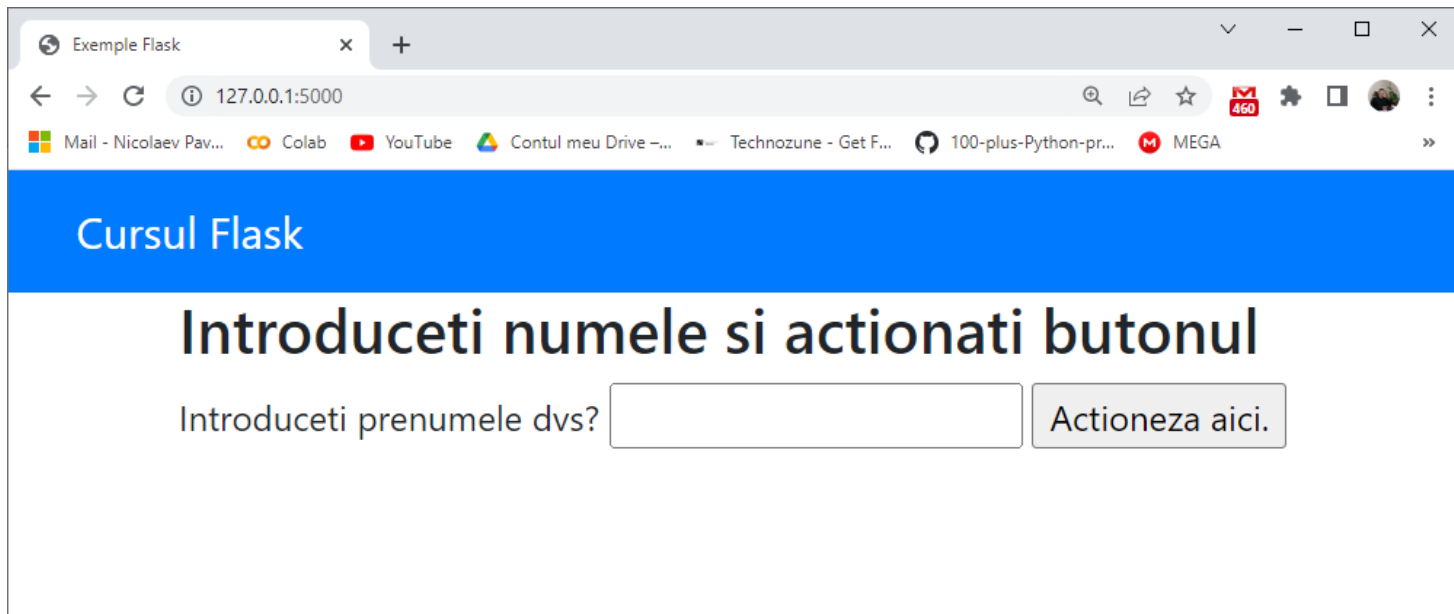
Pagini HTML de pornire

- Se modifică pag1.html care pe lângă formular mai are inclus si câmpul mesajului de alertă când se transmite formularul

```
{% block content %}
<div class="container">
  <h3>Introduceti numele si actionati butonul</h3>
  {% for mess in get_flashed_messages() %}
    <div class="alert alert-warning alert-dismissible fade show" role="alert">
      <strong>Hei {{session['nume']}}!</strong>
      <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
      {{mess}}
    </div>
  {% endfor %}
  <form method="POST">
    {{ form.hidden_tag() }}
    {{ form.nume.label }} {{ form.nume() }}
    {{ form.submit() }}
  </form>
</div>
{% endblock %}
```


Pagina web de pornire

- În browser se introduce adresa `http://127.0.0.1:5000`



Activarea alertei

- Se introduce numele și se actioneaza butonul Actionati aici.

