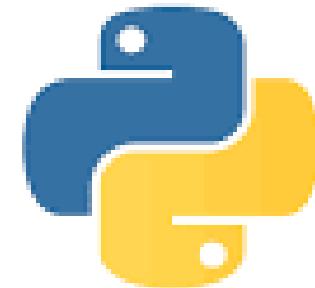


12. Előadás

Python kurzus



Tárgyfelelős:

Dr. Tejfel Máté

Előadó:

Dr. Király Roland

Flask

1. Flask telepítése és elindítása
2. Native code a template fájlokban, Template kezelés
3. Bootstrap a Flask template-ben
4. NAV Bar és Serach Bar – bootstrap
5. GET és POST
6. Oldalak betöltése menüből
7. Statefull vs Stateless, SESSION kezelés és Flash üzenetek



flask

Python webfejlesztés
Flask

LW web framework

Flask alapok

A Flask egy lightweight, Python alapú webes keretrendszer, amely minimalista megközelítést alkalmaz.

Ideális kisebb alkalmazásokhoz vagy prototípusokhoz, de nagyobb projektekhez is.

A Flask támogatja az URL routingot, a sablonkezelést és a RESTful API-k készítését.

Moduláris felépítésének köszönhetően könnyen integrálható bővítményekkel, például adatbázis-kezeléshez vagy hitelesítéshez.

A Flask beépített fejlesztői szervert és hibakeresési eszközöket biztosít a gyors fejlesztés érdekében.

Flask telepítés és elindítása



Flask alapok, telepítés és a webszerver elindítása

Telepítés: pip install flask

<http://127.0.0.1:5000/>: Ez megjeleníti a kezdőoldalt

A <http://127.0.0.1:5000/users/<name>>

Fejlesztői mód bekapcsolása (opcionális):
FLASK_ENV=development

```
from flask import Flask, request, url_for  
from flask import render_template, redirect, session, flash  
from datetime import timedelta
```

Flask alapok, telepítés és a webszerver elindítása

```
from flask import Flask, redirect, url_for  
  
app = Flask(__name__)  
  
@app.route('/')  
def home():  
    return '<h1>Hello Flask</h1>'
```

(indexea.py)

Flask alapok, telepítés és a webszerver elindítása

- Telepíteni kell a Flask csomagot
- Redirect az átirányításokhoz
- url_for a függvények kezeléséhez

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

- ➔ debug mód - nem kell újraindítani a szervert
- ➔ Mint a uvicorn esetén a Fast API-ban.

```
2  
3 app = Flask(__name__)      Debug mode  
4  
5 @app.route('/<user>')  
6 def home(user: str):  
7     #return render_template("index2.html", bo  
8     return render_template("index3.html", nam  
9  
0  
1 if __name__ == '__main__':  
2     app.run(debug=True)
```

HTML file template

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flask App</title>
</head>
<body>
    <h1>Welcome to Flask!</h1>
    <p>This is a basic HTML template served by Flask.</p>
</body>
</html> (simple.html, indexea2.py)
```

Template kezelés

```
from flask import Flask, redirect, url_for, render_template
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

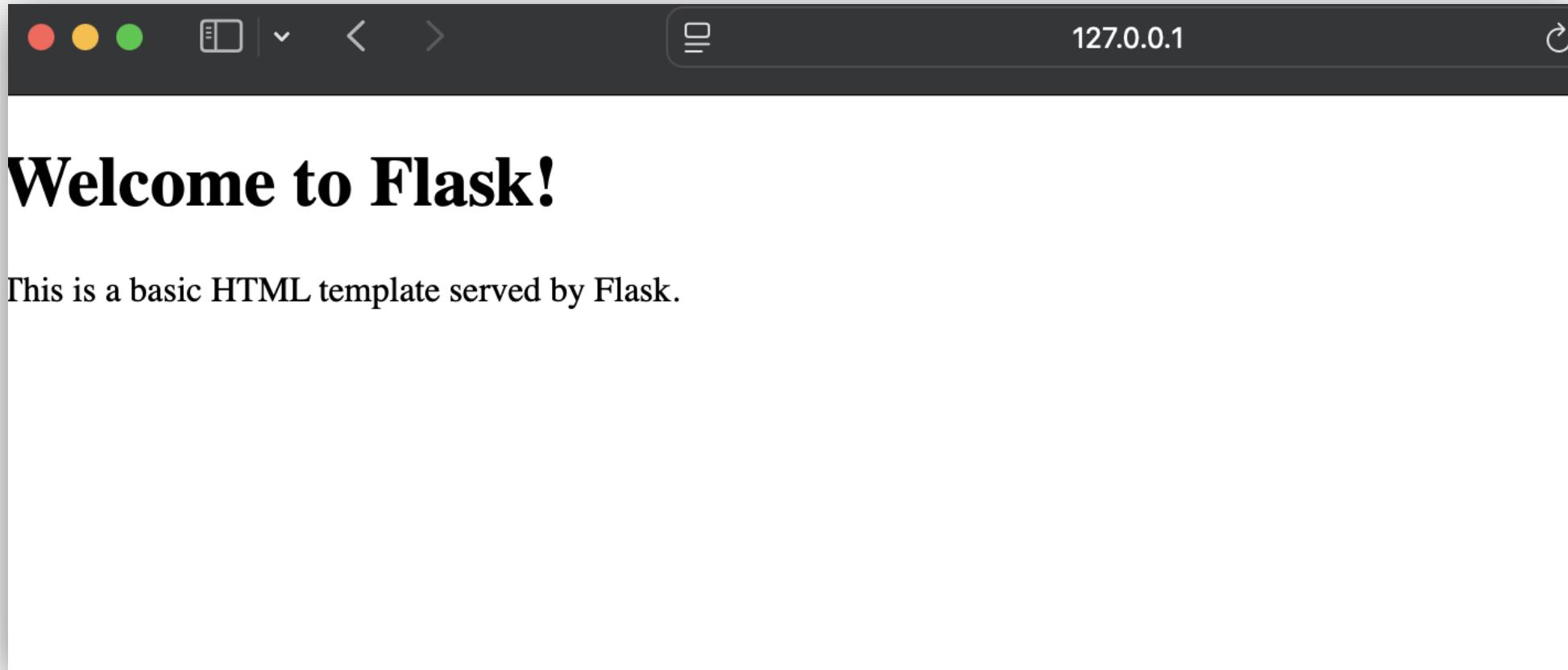
```
def home():
```

```
    return render_template("index.html")
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

Template kezelés – html kimenet



```
1 from flask import Flask, render_template
2
3 app = Flask(__name__)
4
5 @app.route('/<user>')
6 def home(user: str):
7     return render_template("index.html", bodycontent=user)
8
```

Információ átadás (information passing)

```
6      <title>Flask App</title>
7  </head>
8  <body>
9      <h1>Welcome to Flask!</h1>
10     <p>{{bodycontent}}</p>
11 </body>
12 </html>
```

Információ átadás

- {{user}} formában hivatkozunk a Flask tartalomra.
- <https://127.0.0.1:5000/FlaskContentPL>
- (simple2.html, indexea3.py)

```
6      <title>Flask App</title>
7  </head>
8  <body>
9      <h1>Welcome to Flask!</h1>
10     <p>{{bodycontent}}</p>
11 </body>
12 </html>
```



127.0.0.1:5000/FlaskContentPL

Információ
átadás

Welcome to Flask!

FlaskContentPL

```
1  from flask import Flask, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/<user>')
6  def home(user: str):
7      return render_template("index.html", bodycontent=user, message="Logged in")
8
9  if __name__ == '__main__':
10     app.run()
```

Több adat/tartalom is átadható egy renderben.
(simple3.html, indexea3.py)

Native code a template fájlokban



```
</head>
<body>
    <h1>Native python code</h1>
    {% for i in range(0, 10) %}
        <p>Line {{i}}</p>
    {% endfor %}
</body>
```

Native Python kód a HTML template-ben
(index2.html, indexea4.py)

```
<title>Flask App</title>
</head>
<body>
    <h1>Native python code</h1>
    {% for name in namelist %}
        <p>{{name}}</p>
    {% endfor %}
</body>
</html>
```

(indexvars.py, index3.html)

```
app = Flask(__name__)

@app.route('/<user>')
def home(user: str):
    #return render_template("index2.html", bodycontent=user, message="Logged in")
    return render_template("index3.html", namelist=["Név1", "Név2", "Név3"])
```

Listák kezelése

Template kezelés



```
{% extends "template.html" %}  
{% block title %} Home and Navbar later {% endblock %}  
{% block content %}  
Welcome to Flask!  
{% endblock %}
```

(template.html,
index4.html,
indexea5.py)

```
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>{% block title%}{% endblock %}</title>  
</head>  
<body>  
    <div>Menubar</div>  
    <h1>  
        {% block content %}  
        {% endblock %}  
    </h1>  
</body>  
</html>
```

Template

```
{% extends "template.html" %}  
{% block title %} Home and Navbar later {% endblock %}  
{% block content %}  
Welcome to Flask!  
{% endblock %}
```

indexea5.py > ⚙ route

```
1 from flask import Flask, request, url_for, render_template,  
2  
3 app = Flask(__name__)  
4  
5 @app.route('/')  
6 def route():  
7     return render_template("index4.html")  
8  
9 if __name__ == '__main__':  
10    app.run()
```

Bootstrap a Flask template-ben

```
# https://getbootstrap.com  
# https://getbootstrap.com/docs/5.3/components/navbar/#nav
```

Bootstrap

(base.html, index5.html, indexea5.py)

```
{% extends "base.html" %}  
{% block title %}  
    Home and Navbar later  
{% endblock %}  
{% block content %}  
    Flask!  
{% endblock %}
```

A bootstrap link a head tag-be kerül

```
<link  
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.mi  
n.css" rel="stylesheet" integrity="sha384-  
SgOJa3Dml69IUzQ2PVdRZhwQ+dy64/BUtJMjw1MZ8t5HZAp  
cHrRKUc4W0k  
G879m7" crossorigin="anonymous">
```

Java script BSTRP a body tag végére kerül!

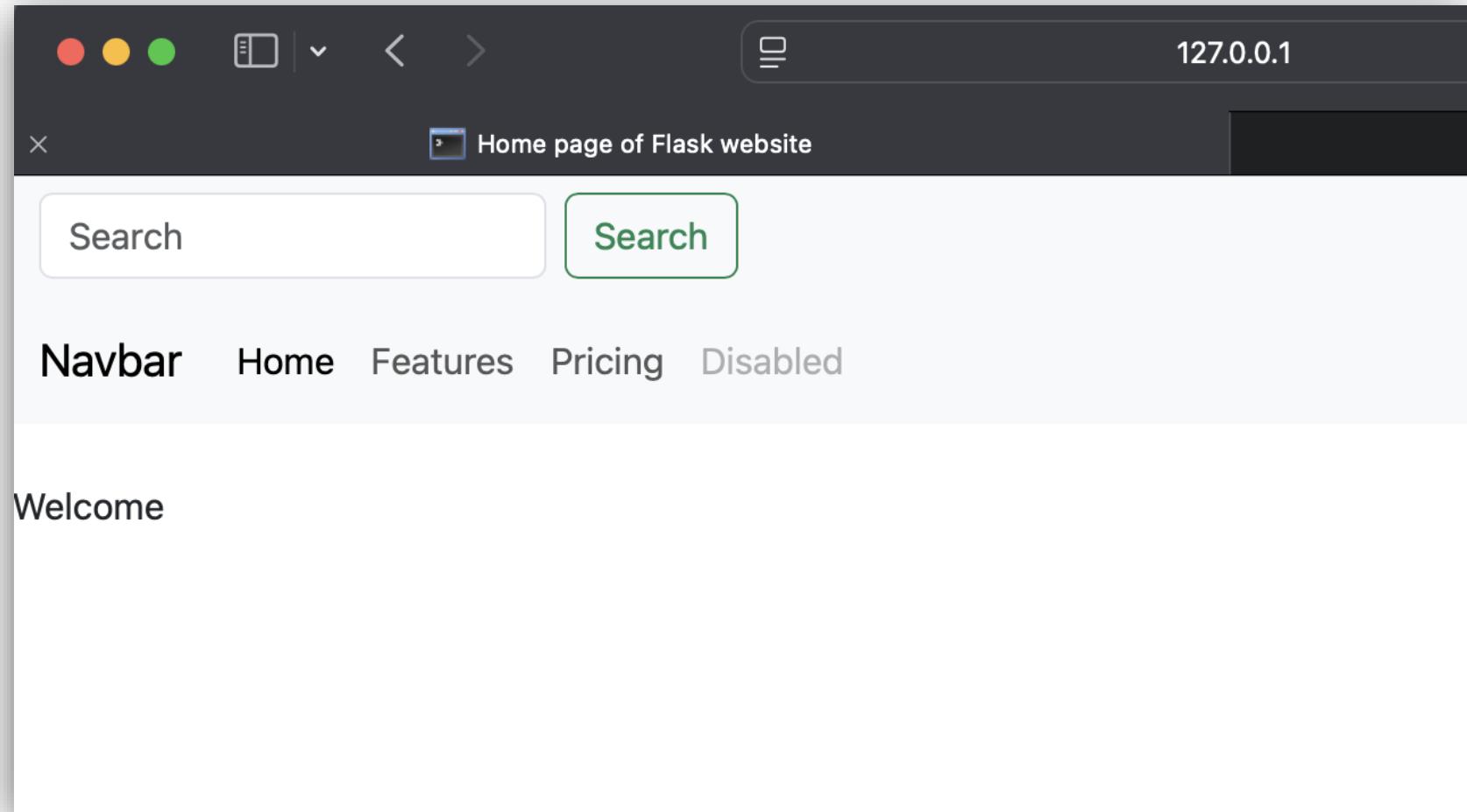
- <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/u
md/popper.min.js" integrity="sha384-
I7E8VWD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2p
M8ODewa9r" crossorigin="anonymous"></script>
- <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstr
ap.min.js" integrity="sha384-
VQqxDN0EQCkWoxt/0vsQvZswzTHUVOImccYmSyhJTp7kGtPed0
Qcx8rK9h9YEgx+" crossorigin="anonymous"></script>

NAV bar és Serach Bar bootstrap-ból



NAV bar

- <https://getbootstrap.com/docs/5.3/components/navbar/#nav>
- (templatenav.html,
index6.html,
indexea6.py)



GET és POST



```
{% extends "templatenav.html" %}  
{% block title %} Login {% endblock %}  
{% block content %}  
<form action="#" method="post">  
    <p><input type="text" name="username"></p>  
    <p><input type="submit" value="submit"></p>  
</form>  
{% endblock %}
```

GET és POST ([login.html](#), [indexlogin.py](#),
[templatenav.html](#))

```
@app.route('/')
def home():
    return render_template("index5.html")

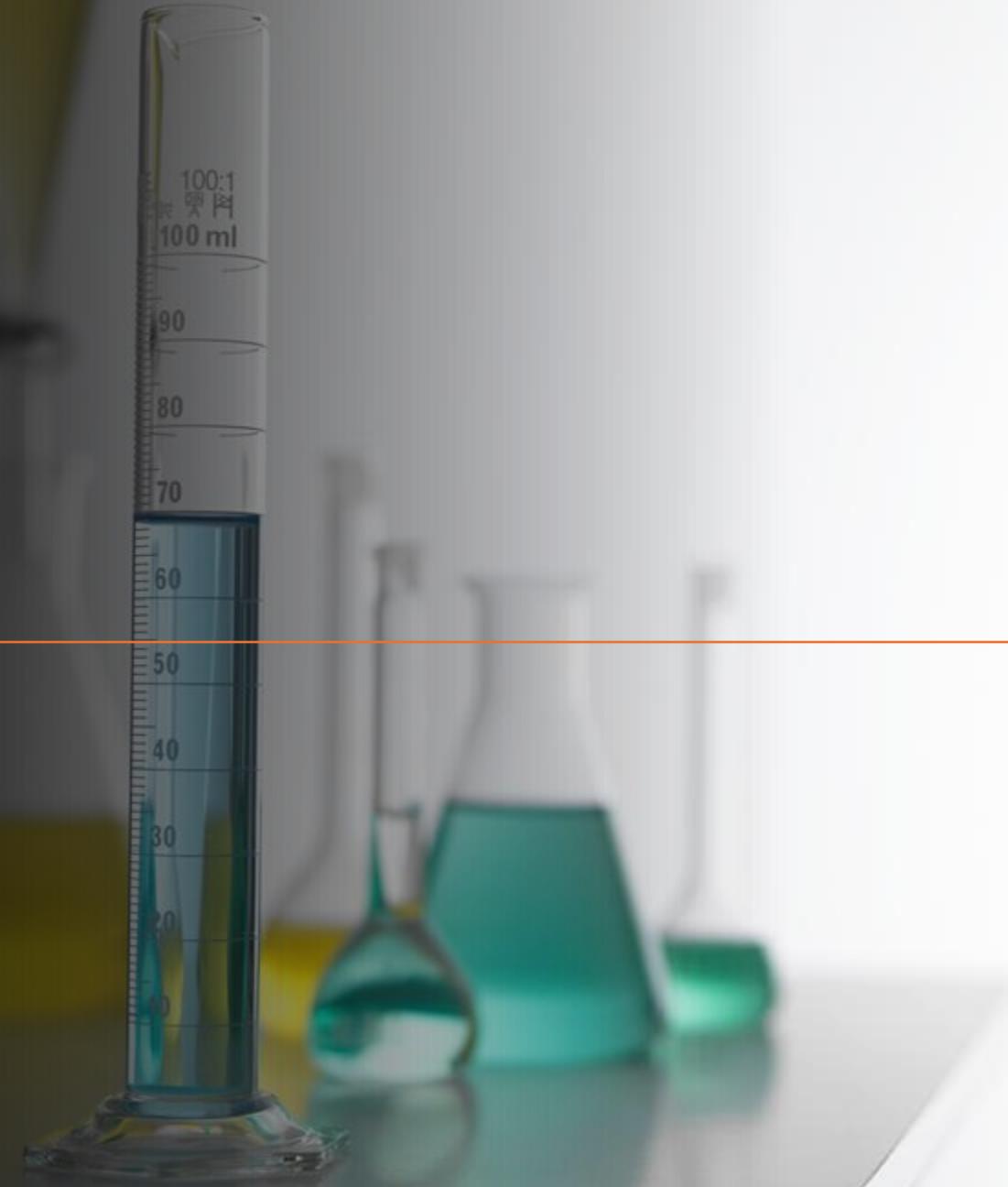
@app.route('/loginpage', methods=["POST", "GET"])
def loginpage():
    if request.method == "POST":
        username = request.form["username"]
        return redirect(url_for("user", usr=username))
    else:
        return render_template("login.html")

@app.route('/<usr>')
def user(usr):
    return f"<h1>{usr}</h1>"
```

GET és
POST

```
127.0.0.1 -- [01/Dec/2025 20:57:00] "GET / HTTP/1.1" 200 -
127.0.0.1 -- [01/Dec/2025 20:57:02] "GET /loginpage
HTTP/1.1" 200 -
127.0.0.1 -- [01/Dec/2025 20:57:10] "POST /loginpage
```

Oldalak betöltése menüből



```
<li class="nav-item">
  <a class="nav-link" href="/loginpage">Login page</a>

</li>
<li class="nav-item">
  <a class="nav-link" href="/otherpage">Otherpage</a>
</li>
```

```
@app.route('/otherpage')
def otherpage():
    return render_template("template.html")
```

Másik oldal betöltése HTML menüből

Statefull vs stateless



Session timeout

```
@app.route('/loginpage', methods=["POST", "GET"])
def loginpage():
    if request.method == "POST":
        session.permanent = True
        username = request.form["username"]
        session["user"] = username
        return redirect(url_for("user"))
    else:
        return render_template("login.html")
```

```
from flask import Flask, request, url_for
from flask import render_template, redirect, session
from datetime import timedelta

app = Flask(__name__)
app.secret_key = 'alma'
app.permanent_session_lifetime = timedelta(minutes=20)
```

Session – kezelés –
állapotok/állapottér
**(indexsession.py,
index5.html,
login.html)**

```
@app.route('/loginpage', methods=["POST", "GET"])
def loginpage():
    if request.method == "POST":
        username = request.form["username"]
        session["user"] = username
        return redirect(url_for("user"))
    else:
        return render_template("login.html")

@app.route('/user')
def user():
    if "user" in session:
        user = session["user"]
        return f"<h1>{user}</h1>"
    else:
        #return f"<h1>no user in session</h1>"
        return redirect(url_for("loginpage"))
```

Flash üzenetek



Flash üzenetek

```
from flask import Flask, request, url_for  
from flask import render_template, redirect, session, flash  
from datetime import timedelta
```

```
@app.route('/logout')  
def logout():  
    session.pop("user", None)  
    ...flash("Logged out!", "Message")  
    #return f"<h1>Logged out</h1>"  
    return redirect(url_for("loginpage"))
```

```
<div class="container-fluid">{% block content %}{%endblock%}</div>
{% with messages = get_flashed_messages() %}
{% if messages %}
<ul class=flashes>
{% for message in messages %}
    <li>{{ message }}</li>
{% endfor %}
</ul>
{% endif %}
{% endwith %}
```

Flash üzenetek kiíratása

Flash üzenetek kiíratása

```
@app.route('/logout')
def logout():
    if "username" in session:
        user = session["username"]
        session.pop("username", None)
        flash("Logout", "Message")
        return redirect(url_for("loginpage"))
    else:
        flash("No user!")
        return redirect(url_for("/"))
```

Flash megjelenítése:

```
{% with messages = get_flashed_messages(with_categories=true) %}  
  {% if messages %}  
    <ul class="flashes">  
      {% for category, msg in messages %}  
        <li class="{{ category|lower|replace(' ', '-') }}>{{ msg }}</li>  
      {% endfor %}  
    </ul>  
  {% endif %}  
{% endwith %}
```

Static folder



static

```
{% extends 'base.html'%}
{% block title %}
    Home Page
{% endblock %}

{% block content %}
    <h1>Home Page</h1>
    <image src="{{ url_for('static', filename='images/image.png')}}">
{% endblock %}
```

```
<html>
<head>
    <title>{% block title %} {% endblock %}</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')}}">
</head>
<body>
    {% block content %}
    {% endblock %}
</body>
</html>
```

Köszönöm a figyelmet!