

CampusLink Workshop — Ümumi Tutorial (Flask • Jinja2 • SQLite • HTML/CSS/Bootstrap)

Bu sənəd yalnız Python bilən, lakin Flask, Jinja2, SQLite və HTML/CSS barədə məlumatı olmayan tələbələr üçün addım-addım bələdçidir. Bütün nümunələr workshop layihəsi — CampusLink — kontekstindədir.

0) Lazımi alətlər və quraşdırma

- Python 3.11+ (və ya mövcud versiyanız)
- Pip, virtual environment (venv)
- Mətn redaktoru (PyCharm Pro tövsiyə olunur)

Virtual mühit və paketlər

Windows

```
python -m venv .venv  
.venv\Scripts\activate  
pip install Flask
```

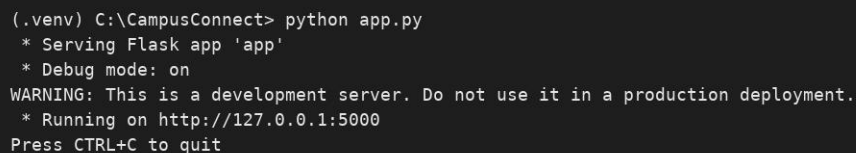
macOS/Linux

```
python3 -m venv .venv  
source .venv/bin/activate  
pip install Flask
```

Flask quraşdırılması üçün əmrlər

Workshop layihəsində artıq requirements.txt var: ``pip install -r requirements.txt``.

Serverin işə düşməsi terminalda təxmini belə görünəcək:



```
(.venv) C:\CampusConnect> python app.py  
* Serving Flask app 'app'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit
```

1) Flask nədir? Route, Request, Response anlayışları

Flask minimal web framework-dür. URL → funksiya xəritəsi qururuq (route). Funksiya HTTP sorğunu qəbul edir və cavab (HTML mətni və ya JSON) qaytarır.

Ən minimal Flask tətbiqi

```
from flask import Flask
```

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

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

```
def hello():
```

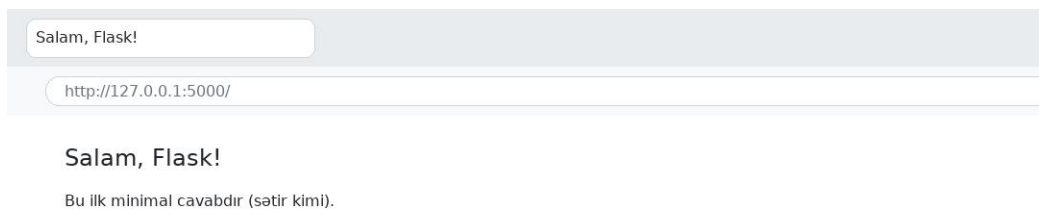
```
    return "Salam, Flask!"
```

```
if __name__ == "__main__":
```

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

Bu faylı `app.py` kimi saxlayıb işə salın: `python app.py`

• `@app.route("/")` — kök URL-ə uyğun funksiya. • `debug=True` — dəyişiklikləri avtomatik yükləyir.



2) Jinja2 şablon sistemi — HTML-i koddan ayırmaq

HTML fayllarını `templates/` qovluğunda saxlayın (içərisində salam.html yaradın). Flask `render_template()` ilə şablonlara (HTML səhifələrinə) məlumat ötürür.

Template render nümunəsi

```
# app.py (parça)
```

```
from flask import Flask, render_template
```

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

```
@app.route("/salam/<ad>")
```

```
def salam(ad):
```

```
    return render_template("salam.html", ad=ad)
```

```
# templates/salam.html
```

```
<!doctype html>
```

```
<html><body>
```

```
<h1>Salam, {{ ad }}!</h1>
```

```
<p>Template içində ad dəyişənindən istifadə edildi</p>
</body></html>
```

Brauzerdə təxmini nəticə:



Base template və bloklar

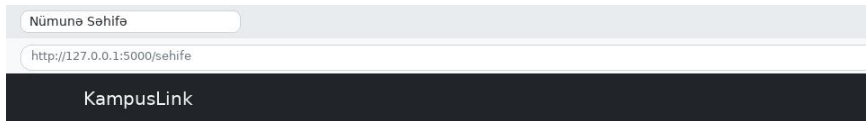
templates/base.html

```
<!doctype html>
<html lang="az">
  <head>
    <meta charset="utf-8">
    <title>{% block title %}KampusLink{% endblock %}</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <nav class="navbar navbar-dark bg-dark">
      <div class="container"><a class="navbar-brand" href="/">KampusLink</a></div>
    </nav>
    <main class="container my-4">
      {% block content %}{% endblock %}
    </main>
  </body>
</html>
```

templates/səhifə.html

```
{% extends "base.html" %}
{% block title %}Nümunə Səhifə{% endblock %}
{% block content %}
  <h1>Bu məzmun base.html üzərinə yazıldı</h1>
{% endblock %}
```

Şablon base: təkrar olunan HTML-i base.html-də saxlayın.



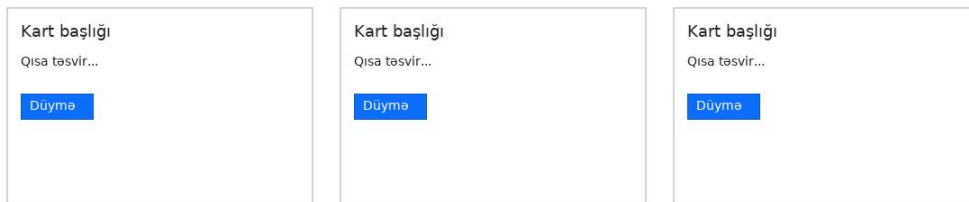
Bu məzmun base.html üzərinə yazıldı

3) HTML/CSS/Bootstrap — sürətli giriş

Bootstrap hazır CSS sinifləri təklif edir: grid sistemi, card, button, form və s. CDN ilə əlavə etmək kifayətdir.

Grid və kart nümunəsi

```
<div class="row g-3">
  <div class="col-md-4">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">Kart başlığı</h5>
        <p class="card-text">Qısa təsvir...</p>
        <a class="btn btn-primary">Düymə</a>
      </div>
    </div>
  </div>
  <!-- daha çox col-md-4 ... -->
</div>
```



Form nümunəsi (POST)

```
<form method="post">
  <div class="mb-3">
    <label class="form-label">Ad</label>
    <input name="name" class="form-control" required>
  </div>
  <button class="btn btn-primary">Göndər</button>
</form>
```

Ad

Göndər

4) SQLite — sadə fayl əsaslı verilənlər bazası

SQLite server tələb etmir, `.db` faylı üzərindən işləyir. Python-da `sqlite3` modulu ilə əlaqə qururuq.

Cədvəl yaratmaq və data daxil etmək

```
import sqlite3
```

```
conn = sqlite3.connect("example.db")  
c = conn.cursor()
```

```
c.execute("""  
CREATE TABLE IF NOT EXISTS posts (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    title TEXT NOT NULL,  
    content TEXT NOT NULL,  
    created_at TEXT NOT NULL  
)""")
```

```
c.execute("INSERT INTO posts (title, content, created_at) VALUES (?, ?, datetime('now'))",  
          ("Birinci yazı", "Salam dünya",))
```

```
conn.commit()  
conn.close()
```

Parametr bağlama (?) SQL injection qarşısını alır.

5) Flask + SQLite integrasiyası (KampusLink konteksti)

`database.py` daxilində bir dəfə bağlantı yaradıb g.db kimi paylaya bilərik. Jinja2-yə Row obyektlərini ötürmək rahatdır.

```
# database.py (sadə nümunə)
```

```
import sqlite3  
from flask import g
```

```
DB_PATH = "campusconnect.db"
```

```
def get_db():  
    if "db" not in g:
```

```

        g.db = sqlite3.connect(DB_PATH)
        g.db.row_factory = sqlite3.Row
        return g.db

def init_db():
    db = sqlite3.connect(DB_PATH)
    db.executescript("""
CREATE TABLE IF NOT EXISTS blog_posts (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    content TEXT NOT NULL,
    created_at TEXT NOT NULL
);""")
    db.commit()
    db.close()

```

Route-da DB istifadə etmək

app.py (parça)

```

from flask import Flask, render_template, request, redirect, url_for
from database import get_db, init_db

```

```

app = Flask(__name__)
init_db()

```

```

@app.route("/blog/")
def blog_list():
    db = get_db()
    posts = db.execute("SELECT * FROM blog_posts ORDER BY id DESC").fetchall()
    return render_template("blog_list.html", posts=posts)

```

```

@app.route("/blog/new", methods=["GET", "POST"])
def blog_new():
    if request.method == "POST":
        title = request.form.get("title", "").strip()
        content = request.form.get("content", "").strip()
        if not title or not content:
            return "Başlıq və məzmun tələb olunur", 400
        db = get_db()
        db.execute("INSERT INTO blog_posts(title,content,created_at) VALUES(?,?,datetime('now'))",
            (title, content))
        db.commit()
        return redirect(url_for('blog_list'))
    return render_template("blog_new.html")

```

6) Formlar, GET/POST, validasiya, flash mesajları

app.py (parça)

```

from flask import flash

```

```

app.secret_key = "demo-secret"

@app.route("/contact", methods=["GET", "POST"])
def contact():
    if request.method == "POST":
        email = request.form.get("email", "").strip()
        message = request.form.get("message", "").strip()
        if not email or not message:
            flash("E-poçt və mesaj mütləqdir.")
            return redirect(url_for("contact"))
        # DB-yə yaz...
        flash("Mesaj göndərildi!")
        return redirect(url_for("contact"))
    return render_template("contact.html")

```

```

# templates/base.html (flash göstərmək)
{% with messages = get_flashed_messages() %}
{% if messages %}
<div class="alert alert-info">{{ messages[0] }}</div>
{% endif %}
{% endwith %}

```

7) Səhifələmə, axtarış və filtr

Səhifələmə üçün LIMIT/OFFSET; axtarış üçün LIKE; filtr üçün WHERE şərtləri.

```

-- SQL
SELECT * FROM blog_posts
WHERE (title LIKE ? OR content LIKE ?)
ORDER BY id DESC
LIMIT ? OFFSET ?;

```

```

# Python (səhifə və axtarış hesabı)
q = request.args.get("q", "").strip()
page = int(request.args.get("page", 1))
per_page = 5
offset = (page-1)*per_page

params = [f"%{q}%", f"%{q}%", per_page, offset]
rows = db.execute("""SELECT * FROM blog_posts
WHERE (title LIKE ? OR content LIKE ?)
ORDER BY id DESC
LIMIT ? OFFSET ?""", params).fetchall()

```

8) Fayl yükləmə (Qalereya nümunəsi)

`enctype="multipart/form-data"` və `request.files` istifadə olunur. Fayl ölçüsü və tipini yoxlayın.

```
# app.py (parça)
```

```
import os, secrets
```

```
from flask import current_app
```

```
ALLOWED = {"png", "jpg", "jpeg", "gif", "webp"}
```

```
def allowed(filename):
```

```
    return "." in filename and filename.rsplit(".", 1)[1].lower() in ALLOWED
```

```
@app.route("/gallery/upload", methods=["GET", "POST"])
```

```
def gallery_upload():
```

```
    if request.method == "POST":
```

```
        file = request.files.get("file")
```

```
        if not file or not allowed(file.filename):
```

```
            flash("Şəkil faylı tələb olunur.")
```

```
            return redirect(url_for("gallery_upload"))
```

```
        ext = file.filename.rsplit(".", 1)[1].lower()
```

```
        name = f"{secrets.token_hex(8)}.{ext}"
```

```
        path = os.path.join(current_app.root_path, "static", "uploads", name)
```

```
        file.save(path)
```

```
        # DB-yə filename daxil edin...
```

```
        return redirect(url_for("gallery_list"))
```

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

```
<!-- templates/gallery_upload.html -->
```

```
<form method="post" enctype="multipart/form-data">
```

```
    <input type="file" name="file" class="form-control" accept="image/*" required>
```

```
    <button class="btn btn-primary mt-2">Yüklə</button>
```

```
</form>
```

9) Sorğular (Polls) — JSON seçimlər və sessiya ilə məhdudiyyət

Seçimləri JSON kimi saxlayırıq; səs verildikdən sonra sessiyada işarə qoyuruq ki, təkrar səs verilməsin (sadə demo).

```
# poll cədvəli (sadə)
```

```
CREATE TABLE polls(
```

```
    id INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
    question TEXT NOT NULL,
```

```
    options_json TEXT NOT NULL,
```

```
    is_closed INTEGER NOT NULL DEFAULT 0
```

```
);
```

```
# Flask (parça)
```

```
from flask import session
```

```
import json, datetime
```

```
@app.route("/polls/<int:pid>", methods=["GET", "POST"])
```

```
def poll_detail(pid):
```



```

poll = db.execute("SELECT * FROM polls WHERE id=?", (pid,)).fetchone()
options = json.loads(poll["options_json"])
if request.method == "POST" and not poll["is_closed"]:
    if session.get(f"voted_{pid}"):
        return redirect(url_for("poll_detail", pid=pid))
    idx = int(request.form.get("option_index", "-1"))
    if 0 <= idx < len(options):
        db.execute("INSERT INTO poll_votes(poll_id, option_index, created_at)
VALUES(?,?,datetime('now'))", (pid, idx))
        db.commit()
        session[f"voted_{pid}"] = True
        return redirect(url_for("poll_detail", pid=pid))
# nəticələri hesabla...
return render_template("poll_detail.html", poll=poll, options=options)

```

10) Təhlükəsizlik və yaxşı praktikalar (qısa)

- Parametr bağlama istifadə edin (`?`), SQL-ə birləşdirmə etməyin.
- Admin əməliyyatlarını ən azı parolla qoruyun (dərsdə demo), idealda login sistemi.
- `SECRET_KEY`-i mühit dəyişəni ilə saxlayın.
- `debug=True` yalnız inkişaf (development) zamanı.

11) Layihənin işə salınması və struktur

```

# İşə salma
pip install -r requirements.txt
python app.py
# http://127.0.0.1:5000 açın

```

Struktur (qısa): app.py, database.py, modul faylları (blog.py və s.), templates/, static/ (CSS, uploads/).

12) Test və yoxlama siyahısı (Checklist)

- Hər modulun əsas səhifə və detallar səhifəsi açılır.
- Formlar POST edir, məcburi sahələrdə validasiya var.
- DB-də yeni sətirlər yaranır, səhifələrdə görünür.
- Axtarış və filtr işləyir.
- Yüklənən fayllar `static/uploads/`-da saxlanır və görünür.
- Sorğuda səs verib nəticələri görmək olur.

- Navbar linkləri bütün modullara aparır.

13) Screenshot/GIF yerləri (yaradılacaq platforma haqqında təsəvvür)

13.0) Ana səhifə

KampusLink Blog Tədbirlər Forum Qalereya Sorğular Əlaqə

KampusLink — Genişləndirilmiş versiya

Flask + SQLite + Jinja2 ilə qurulmuş icma platforması. Modul başına daha çox funksiya!

- Blog: axtarış, teq filtr, düzəliş/silmə, səhifələmə
- Tədbirlər: kapasite, təkrar qeydiyyatın qarşısı, CSV ixrac
- Forum: axtarış, bəyənmə, pin, səhifələmə
- Qalereya: detal, filtr, admin silmə, ölçü limiti
- Sorğular: yeni sorğu, bağlama/açma, sessiya ilə anti-dup səs
- Əlaqə: status dəyiş, filtr, CSV ixrac

13.1) Blog

KampusLink Blog Tədbirlər Forum Qalereya Sorğular Əlaqə

Blog

Yeni yazı

Hamısı

Filtirlə

[KampusLink-ə xoş gəldiniz](#)

2025-08-15 20:52 — teqlər: intro,xəber

Workshop-da birlikdə qurduğumuz icma platforması!

Oxu Düzəlt Sil

[3 saata necə qurduq](#)

2025-08-15 20:52 — teqlər: flask,workshop

Flask + SQLite + Jinja2 + komanda işi.

Oxu Düzəlt Sil

[Daxili işlər](#) Draft

2025-08-15 20:52 — teqlər: draft

3 saata necə qurduq

2025-08-15 20:52 — teqlər: flask,workshop

Flask + SQLite + Jinja2 + komanda işi.

Geri

13.2) Tədbirlər

Tədbirlər

Yeni tədbir

Python Study Jam

A12 otağı — Qalan yer: 29/30

2025-09-10 18:00

Data Viz Gecəsi

Lab 3 — Qalan yer: 20/20

2025-09-15 19:00

Python Study Jam

A12 otağı | 2025-09-10 18:00

Birlikdə Python məsələləri həll edirik.

Qeydiyyatları CSV kimi endir

Qeydiyyat

Adınız

E-poçt

Qeydiyyatdan keç

Qeydiyyat siyahısı

- Fara (hferid23@gmail.com) — 2025-08-15 21:47

Öz qeydiyyatlarınızı yoxla: </events/my-registrations?email=you@example.com>

Yeni tədbir

Admin parolu

Başlıq

Tarix (YYYY-MM-DD HH:MM)

Məkan

Təsvir

Kapasite

100

Yarat

13.3) Forum

Forum

Yeni mövzu

Axtar...

Axtar

PIN Ən yaxşı Python fəndləri?	Aysu — 2025-08-15 20:52
Likes: 5	
Uyğun büdcəli noutbuk?	Murad — 2025-08-15 20:52
Likes: 2	

Ən yaxşı Python fəndləri?

Aysu — 2025-08-15 20:52

Sevdiyiniz Python fəndlərini paylaşın!

Beyən (5) Pin

Cavablar

Kamran: List comprehension və f-strings çox işə yararır ;) — 2025-08-15 20:52

Cavab yaz

Adınız

Cavabınız

Göndər


13.4) Qalereya

KampusLink [Blog](#) [Tədbirlər](#) [Forum](#) [Qalereya](#) [Sorğular](#) [Əlaqə](#)


Qalereya

[Yüklə](#)

[Filtrlə](#)



Kampus Günbatımı
Admin — 2025-08-15 20:52



Workshop Komandası
Admin — 2025-08-15 20:52

KampusLink [Blog](#) [Tədbirlər](#) [Forum](#) [Qalereya](#) [Sorğular](#) [Əlaqə](#)

Şəkil yüklə

Başlıq

Adınız

Şəkil faylı

[Choose File](#) no file selected

[Yüklə](#)

13.5) Sorğular

KampusLink [Blog](#) [Tədbirlər](#) [Forum](#) [Qalereya](#) [Sorğular](#) [Əlaqə](#)

Sorğular

[Yeni sorğu](#)

[Mentor sistemi istəyirsiniz?](#)[Bağlı](#)

[KampusLink mövzusu?](#)[Açıq](#)

KampusLink mövzusu?

Bağla

- ☐ Tünd
☐ Açıq
☐ Auto

Səs ver

Nəticələr (4 səs)

Tünd: 2 (50.0%)

Açıq: 1 (25.0%)

Auto: 1 (25.0%)

Mentor sistemi istəyirsiniz?

Aç

Sorğu bağlıdır, səs vermək mümkün deyil.

Nəticələr (1 səs)

Bəli: 0 (0.0%)

Xeyr: 1 (100.0%)

Yeni sorğu

Admin parolu

Sual

Seçimlər (hər sətirdə bir seçim)

Yarat

13.6) Əlaqə

Əlaqə / Geri bildiriş

Adınız

E-poçt

Kategoriya (məs., bug, idea)

Mesajınız

Göndər

Admin baxış: /admin/feedback?password=admin123