

Flask'ın Temelleri ve CSV Verilerinin Web Uygulamalarında Kullanımı

MELİS YANIK

Günümüz dünyasında artık yazılım sektörü deyince akıllara yapay zeka, veri analizi ve makine öğrenmesi gelmektedir. Bu noktada odak en etkili ve doğruluğu yüksek modeli geliştirmek, veriyi anlamlı bir hale getirmek olsa da, bu verilerin kullanıcılara hangi formatta ve nasıl sunulduğu da önem arz etmektedir. Kullanıcıların anlayabileceği, kolay erişilebilen bir web ara yüzü çözüm olarak ele alınabilir.

Veri analizi konusunda sahip olduğu zengin kütüphane yapısından dolayı Python, bu alandaki en popüler programlama dilidir. Tensorflow, Scikit-learn, Numpy, Keras, PyTorch, Matplotlib, Pandas bu kütüphanelerin en çok kullanılanlarındandır. Fakat bu kütüphaneler kullanılarak yapılan projelerin, canlı ortamda gösterilmesi için bir web framework arayışı doğmuştur. Flask, Python ile yazılmış mikro düzeyde bir web uygulama çatısıdır. Bu yazıda Flask hakkında bilgiler ve CSV formatındaki bir dosyanın Flask ile web arayüzünde gösterilmesi hakkında bilgiler edineceksiniz.

1-Flask Nedir?

Son yıllarda Django ile beraber Python için en popüler web çerçevesi olarak yer alan Flask'ın mikro web çatısı olması, temel düzeyde, tam teşekküllü olmadan çalışmasından kaynaklanır. Veri tabanı soyutlama ya da form doğrulama gibi özellikleri yoktur. Fakat sonradan eklenebilen ve kendi özelliğiymiş gibi davranabilen uzantıları sayesinde, bir web çatısından beklenen her şeyi yapar duruma gelebilmektedir. Ölçeklenebilir ve hafif yapısı sayesinde kısa sürede popüler olmuştur. Küçük ve orta büyüklükteki projeler için ise idealdir. Aşağıda Flask'ın diğer framework yapıları ile karşılaştırılması verilmiştir:

Özellik	Flask	Django	FastAPI
Tür	Micro Framework	Full Stack Framework	API Framework
Öğrenme Kolaylığı	Yüksek	Orta	Yüksek
Performans	Yüksek	Orta	Çok Yüksek
ORM	Harici	Dahili	Harici
Admin Paneli	Yok	Var	Yok
API Geliştirme	İyi	İyi	Çok İyi
Veri Bilimi Projeleri	Çok Uygun	Uygun	Çok Uygun
Kullanım Senaryosu	Küçük-Orta Projeler	Büyük Kurumsal Projeler	Modern API ve Mikroservisler

Flask'ın diğer framework teknolojileriyle karşılaştırılması

Ayrıca Flask, SQL ve NoSQL ile de kolayca entegrasyon sağlayabilir. ORM desteği yoktur fakat özelleştirilebilir olmasıyla ve dokümantasyonunun oluşuyla da geliştiriciler sıkça tercih etmektedir.

2- Flask İle Neler Yapılabilir?

- Makine öğrenmesi ve veri bilimi
- RESTful Api geliştirme
- Dinamik web uygulamaları
- Veri odaklı araçlar
- Prototip ve MVP (Minimum Viable Product) Geliştirme
- Mikro Servis Mimarileri

3-Flask Kullanan Büyük Şirketler

Büyük ölçekli şirketlerin birçoğu Flask kullanmaktadır. Bazıları birincil olarak kullanmasa da, ek geliştirmelerde destek almaktadırlar.

- Samsung
- Netflix
- Uber
- Zalando
- Trivago
- Reddit
- Pinterest
- Patreon
- Zomato

4- Flask'ın Yapısı ve Örnek Üzerinden Anlatımı

Flask'ı projelerinize dahil etmek gayet kolaydır. Projenizin ihtiyacına göre kodlar değişse de sabit kalan bazı yapılar vardır. Aşağıdaki kod satırında:

```
from flask import Flask, render_template, request, jsonify, send_from_directory
```

Flask'ın projeye dahil edilmesi

import komutu projenize Flask eklemeniz için yeterlidir. Render_template ise kullanıcılara HTML sayfasını göstermek için kullanılmaktadır. Request kullanıcıdan veri alırken, jsonify komutu ise python verilerini json formatına çevirmektedir. send_from_directory ise belirli klasördeki dosyaları göndermektedir.

Uygulamanın başlatıldığı nokta:

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

Uygulamanın çalışmaya başladığı nokta

Oldukça sade ve anlaşılır bir yapıda olan Flask, dinamik yönlendirme özelliğine sahiptir. Örneğin:

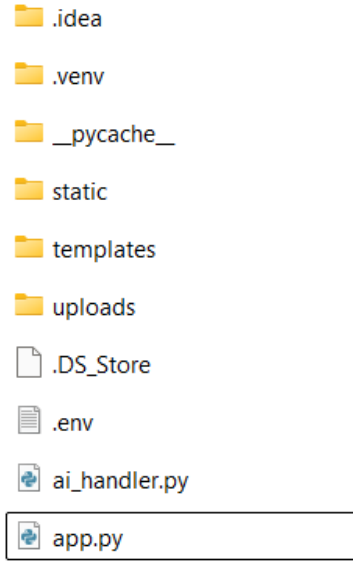
```
@app.route('/')
def home():
    return render_template('index.html')
```

app.py dosyasının yönlendirme kodları

Kullanıcı siteye girdiğinde Flask bu isteği alır ve hangi sayfanın karşılık geleceğini bilip tarayıcıda gösterir. Yukarıdaki kod örneğinde index sayfasına yönlendirme yapılmaktadır.

Tüm bu kodlar `app.py` dosyası içerisinde. Uygulamanın bel kemiği olan bu dosya, aynı zamanda programın da başlatıldığı yerdir.

Yapay zeka entegrasyonu olan bir web uygulamasının dosya yapısı şu şekildedir:



app.py dosya görünümü

Görüldüğü üzere tüm uygulama [app.py](#) dosyasıyla çalışmaktadır. ai_handler.py ise yapay zeka entegrasyonu için gereken dosyadır.

5- Flask ve Asp .NET Arasındaki Farklar

Yazının bu bölümünde iki çok popüler arka uç teknolojisi olan Flask ve .NET arasındaki farkları inceleyeceğiz. .NET büyük ölçekli projeler için idealdir fakat Flask her ne kadar uzantılarla daha büyük ölçeklere genişletilse de, bir noktada .NET kapsamında değildir.

.NET; LINQ, Generics, async/await gibi yerleşik özellikler sunar, buna karşın Flask kütüphane ekleme sistemi ile çalışır. Geliştirme ortamları da tamamen farklıdır. Hız konusunda ise Flask hafif yapısından dolayı öne çıkmaktadır. Aşağıda birkaç kod örneğiyle iki teknoloji arasındaki farklar ele alınmıştır:

```
/*Asp.NET*/
using Microsoft.AspNetCore.Mvc;

public class HomeController : Controller
{
    public IActionResult Index()
    {
        return Content("Merhaba ASP.NET Core");
    }
}
```

Kullanıcı HomeController üzerinden Index sayfasına erişebilmektedir.

```
#Flask
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "Merhaba Flask"
```

Burada da aynı durum vardır fakat yüklenme şekli farklıdır. Kullanıcı <http://localhost:5000/> adresini tarayıcıya girer, varsayılan adres yüklenir.

Bir başka örnek olarak kullanıcı adını ekrana yazdırmayı alalım:

```
/*Asp .NET*/
public IActionResult User(string name)
{
    return Content($"Hoş geldin {name}");
}
```

Controller içinde ActionResult kullanarak kullanıcı adının sayfada gözükmesini sağladık.

Fakat Flask için bu yapı daha farklı:

```
#Flask
@app.route("/user/<name>")
def user(name):
    return f"Hoş geldin {name}"
```

Ürün detay sayfasında numarası seçilen bir ürünün gösterilmesi:

```
/*Asp .NET*/
public IActionResult Detail(int id)
{
    return Content($"Ürün No: {id}");
}#Flask
@app.route("/product/<int:id>")
def product(id):
    return f"Ürün No: {id}"
```

Burada id değerini özellikle integer olarak belirttik çünkü Flask URL rotalarındaki dinamik değişkenleri default şekilde string olarak alır.

Bir index sayfasında liste göstermek istediğimizde:

```
/*Asp .NET Core*/
public IActionResult Index()
{
    var students = new List<string>
    {
        "Ali",
        "Veli",
        "Ayşe"
    };

    return View(students);
}<!--index.html-->
<ul>
@foreach(var student in Model)
{
    <li>@student</li>
}
</ul>
```

Flask'ta mantık aynı olsa da:

```
#Flask
@app.route("/")
def home():
    students = ["Ali", "Veli", "Ayşe"]

    return render_template(
        "index.html",
        students=students
    )<!--index.html-->
<ul>
{% for student in students %}
    <li>{{ student }}</li>
{% endfor %}
</ul>
```

6- Flask İle Verilerin Web Ortamına Çekilmesi

Yazımızda son olarak Flask için oldukça önemli bir konuya değineceğiz. Comma-Separated Value (CSV) dosyaları, Excel'de oluşturulabilen ve düzenlenebilen bir dosya formatıdır. Tablo görünümündeki veriler, CSV olarak adlandırılabilir. E-ticaret ve veri transferi alanlarında kullanılan bu dosya formatının, Flask çerçevesinde de en çok kullanılan formatlardan biri olduğunu söyleyebiliriz. CSV dosyaları Python ile Pandas gibi kütüphanelerle okunabilir hale gelmektedir. Elde edilen verilerin kullanıcılara farklı şekillerde sunulması gerektiğinde bu köprüyü yine Flask kurmaktadır. Örneğin `kisiler.csv` adında bir dosyamız olsun ve bunu web ortamında göstermek isteyelim:

	A	B	
1	Ad	Yaş	
2	Ali	25	
3	Ayşe	30	
4	Mehmet	22	
5			
6			

kisiler.csv

Ardından `app.py` dosyası açarak Flask kütüphanesini importladık. CSV dosyasından alabilmesi için pandas kütüphanesini de ekledik. Uygulama başlatılır, CSV dosyası pandas kütüphanesi ile okunur, tüm satırlar tek tek gezilir ve incelenir, veriler ekrana yazılır. Son olarak da tarayıcıya gönderilir.

```
from flask import Flask
import pandas as pd

app = Flask(__name__)

@app.route("/")
def home():
    try:
        df = pd.read_csv("kisiler.csv")
        sonuc = ""
        for index, row in df.iterrows():
            sonuc += f"{row['Ad']} - {row['Yas']}<br>"
        return sonuc
    except Exception as e:
        return f"Hata: {str(e)}"

if __name__ == "__main__":
    app.run(debug=True, port=5005)
```

app.py dosyasının kodları

Uygulamanın çalıştırılmış hali:

Ali - 25
Ayşe - 30
Mehmet - 22

kisiler.csv dosyasının web ortamında gösterilmesi

Görüldüğü üzere Flask, az kodla ve kütüphaneleriyle daha uzun sürecek işlemleri tek satırda yapmakta ve kullanıcılara kolaylık sağlamaktadır. Popüler olmasının ardındaki diğer sebeplerden biri ise budur. Aynı işlemler farklı platformlarda daha fazla kod gerektirmektedir.

KAYNAKÇA

1. Flask Documentation—Quickstart
<https://flask.palletsprojects.com/en/stable/quickstart/>
2. Werkzeug Documentation—Routing
<https://werkzeug.palletsprojects.com/en/stable/routing/>
3. GeeksforGeeks—Uploading and Reading a CSV File in Flask
<https://www.geeksforgeeks.org/python/uploading-and-reading-a-csv-file-in-flask/>
4. Microsoft Support—Create or Edit CSV Files to Import into Outlook
<https://support.microsoft.com/tr-tr/outlook/create-or-edit-csv-files-to-import-into-outlook>
5. Flask Rehber—Flask Neden Microframework Olarak Geçiyor?
<https://github.com/erkamesen/Flask-Rehber#flask-neden-microframework-olarak-ge%C3%A7iyor->
6. StackShare—C# vs Flask
<https://stackshare.io/stackups/c-sharp-vs-flask>
7. Mobilhanem—Flask Eğitimi
<https://www.mobilhanem.com/flask-egitimi/>
8. Career Karma—Companies That Use Flask
<https://careerkarma.com/blog/companies-that-use-flask/>
9. Buğra Kaya—Flask ve En Çok Kullanılan Kütüphaneleri
<https://medium.com/@cbugrakaya/flask-ve-en-%C3%A7ok-kullan%C4%B1lan-k%C3%BCt%C3%BCphaneleri-907e07386553>
10. Dev.to—5 Flask Project Ideas and Why You Should Implement Them
<https://dev.to/heyaliona/5-flask-project-ideas-and-why-you-should-implement-them-4n51>
11. Coderspace—En Çok Kullanılan 10 Python Frameworkü
<https://coderspace.io/blog/en-cok-kullanilan-10-python-frameworku/>
12. Coderspace Sözlük—Flask Nedir?
<https://coderspace.io/sozluk/flask-nedir/>
13. FastAPI Documentation
<https://fastapi.tiangolo.com/>
14. Techcareer Sözlük—Flask
<https://www.techcareer.net/dictionary/flask>
15. AWS—What is Python?
<https://aws.amazon.com/tr/what-is/python/>