



## Katkıda Bulunanların Listesi

- Ahmet Berkay Avan
- Mehmet Begun

## Task Matrix

Görev	Atanan Kişi
RSS haber çekme modülü (rss_scraper.py)	Ali Hakan Kıncal
Metin temizleme ve NLP (text_cleaner.py)	Barış Emre Ahi
Makine öğrenmesi kategorilendirme (lda_categorizer.py)	Mehmet Begun
SQLite veritabanı yönetimi (db_helper.py)	Ahmet Berkay Avan
Flask web arayüzü (app.py, index.html)	Ali Hakan Kıncal, Barış Emre Ahi
Öneri modülü (suggestions.py)	Mehmet Begun
Dokümantasyon (README.md, Design Document)	Ahmet Berkay Avan

## **İçindekiler**

1. Giriş (Proje Tanıtımı)	2
2. Sistem Mimarisi	2
3. Teknoloji ve Araçlar	3
4. Uygulama Detayları	3
5. Çalışma Akışı (Workflow)	4
6. Kurulum ve Çalıştırma	4
7. Kullanım Senaryoları	5
8 Teknoloji Seçimleri ve Karşılaştırmalar	5
9. Sonuç ve Gelecek Geliştirmeler	5
10. Referanslar	6

# İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi

## 1. Giriş (Proje Tanıtımı)

### 1.1 Projenin Amacı

Bu proje, iklim krizi ile ilgili haberleri analiz eden ve yapay zeka destekli öneriler üreten bir sistem geliştirmeyi amaçlamaktadır. Sistem, çevrimiçi kaynaklardan RSS beslemeleri aracılığıyla veri toplayacak, doğal dil işleme (NLP) teknikleri ile analiz edecek ve kullanıcıya çevre dostu öneriler sunacaktır.

## 2. Sistem Mimarisi

Sistem, katmanlı mimari (layered architecture) prensibine göre tasarlanmıştır:

### 2.1 Veri Toplama Katmanı

- Çevrimiçi kaynaklardan (RSS) haber verilerini çeker.
- HTML etiketlerini kaldırarak ve gereksiz kelimeleri temizleyerek verileri işlenebilir hale getirir.

### 2.2 Analiz Katmanı (NLP İşleme)

- Doğal dil işleme (NLP) algoritmalarını kullanarak haber içeriklerini analiz eder.
- Önemli anahtar kelimeleri belirler.
- Latent Dirichlet Allocation (LDA) algoritması ile haberleri otomatik olarak kategorilere ayırır.

### 2.3 Öneri Motoru

- Anahtar kelimelere dayalı çevre dostu öneriler oluşturur.
- Önerileri kullanıcıya anlık olarak sunmaktadır..

### 2.4 Kullanıcı Arayüzü (Frontend)

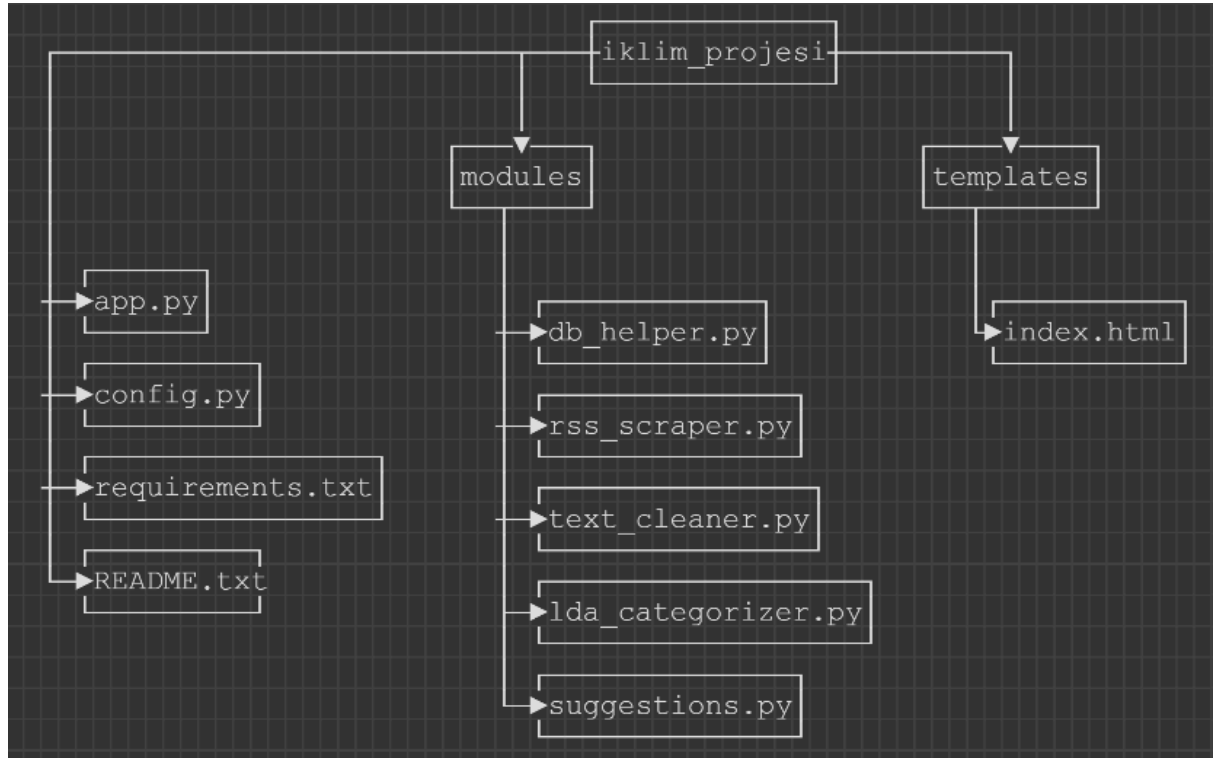
- Haberleri ve önerileri gösteren bir web arayüzü sağlar. (Flask + HTML)
- Haberleri kategoriye göre filtreleme imkanı sunar.

### 3. Teknoloji ve Araçlar

Backend	Python (Flask)
Veritabanı	SQLite
Doğal Dil İşleme (NLP)	NLTK, BeautifulSoup
Makine Öğrenmesi	Latent Dirichlet Allocation (LDA), Scikit-learn
Özellik Çıkarma	CountVectorizer (Bag of Words - BoW)
RSS Veri Çekme	Feedparser
Frontend	Flask + Jinja Templating + HTML
Dağıtım (Deployment)	Flask

### 4. Uygulama Detayları

#### 4.1 Kod Yapısı



## 4.2 Ana Modüller ve Görevleri

Modül	Görevi
app.py	Flask web sunucusunu çalıştırır ve arayüzü yönetir.
modules/db_helper.py	SQLite veritabanını yönetir, haberleri saklar.
modules/rss_scraper.py	RSS kaynaklarından haberleri çeker ve işler.
modules/text_cleaner.py	Metinleri temizler ve NLP ön işleme yapar.
modules/lda_categorizer.py	Makine öğrenmesi (LDA) ile haberleri kategorilere ayırır.
modules/structures.py	Kategori etiketlerini (topic_labels) ve aksiyon önerilerini (action_suggestions) yönetir.

## 5. Çalışma Akışı (Workflow)

1. Uygulama başlatılır (app.py çalıştırılır).
2. Veritabanı başlatılır (db\_helper.py).
3. RSS haberleri çekilir (rss\_scraper.py).
4. Metinler temizlenir (text\_cleaner.py).
5. LDA modeliyle haberler kategorilere ayrılır (lda\_categorizer.py).
6. Haberler veritabanına kaydedilir.
7. Flask, haberleri HTML şablonu ile kullanıcıya sunar.

## 6. Kurulum ve Çalıştırma

- **Gerekli Bağımlılıkları Yükle:**

```
pip install -r requirements.txt
```

- **Uygulamayı Başlat:**

```
python app.py
```

- **Web Arayüzüne Erişim**

Uygulama çalıştıktan sonra tarayıcıdan <http://127.0.0.1:5000/> adresine giderek iklimle ilgili haberleri ve önerileri görüntüleyebilirsiniz.

## 7. Kullanım Senaryoları

Use Case	Fonksiyonel Gereksinim
Veri toplama	RSS'den veri çekebilmeli
NLP analizi	Metinleri işleyip kategorilere ayırmalı
Öneri üretme	Kullanıcılara çevre dostu öneriler sunmalı
Raporlama	PDF/CSV rapor oluşturmali

**Veri Akışı:** Kaynak → Veri toplama → Analiz → Öneri oluşturma → Kullanıcı

**Durum Değişimleri:** Haberlerin analiz edilmesi, yapay zeka ile işlenmesi ve önerilere dönüşümü.

## 8. Teknoloji Seçimleri ve Karşılaştırmalar

Teknoloji	Alternatif	Karşılaştırma
Flask	Django	Flask, küçük projelerde daha hızlı geliştirme sağlar.
SQLite	PostgreSQL	SQLite, hafif ve basit olması nedeniyle tercih edildi.
spaCy	NLTK	spaCy daha hızlı ve modern NLP teknikleri sunuyor.

## 9. Sonuç ve Gelecek Geliştirmeler

Bu proje, çevresel haberleri analiz ederek kullanıcıların bilinçli kararlar almasına yardımcı olacak modüler ve ölçeklenebilir bir yapıya sahiptir. Geliştirilmesi kolaydır ve yeni teknolojilerle entegre edilebilir.

### ✓ Mevcut Sınırlamalar:

- Daha fazla haber kaynağı eklenebilir.
- Daha gelişmiş bir öneri motoru kullanılabilir.
- Kullanıcı arayüzü daha kullanışlı hale getirilebilir.
- Daha çok topic label ve action suggestion eklenebilir.

### ✓ Gelecekteki Geliştirmeler:

- Daha fazla haber kaynağı (API entegrasyonu).
- Mobil uygulama entegrasyonu.

## 10. Referanslar

- Flask: <https://flask.palletsprojects.com/en/stable/>
- NLTK: <https://www.nltk.org/>
- Scikit-learn (LDA):  
<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html>
- BeautifulSoup: <https://www.crummy.com/software/BeautifulSoup/>