



## İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi

### Katkıda Bulunanların Listesi

- Ahmet Berkay Avan
- Mehmet Begun
- Ali Hakan Kincal
- Barış Emre Ahi

### Task Matrix

Görev	Atanan Kişi
Giriş	Mehmet Begun
Deployment Overview	Barış Emre Ahi
Deployment Process	Ali Hakan Kincal
Configuration Plan	Ahmet Berkay Avan
Belgelerin Son Düzenleme ve Formatlaması	Ahmet Berkay Avan, Barış Emre Ahi, Ali Hakan Kincal, Mehmet Begun

### İÇİNDEKİLER

<b>Katkıda Bulunanların Listesi</b>	<b>1</b>
<b>1. Giriş</b>	<b>2</b>
<b>2. Deployment Overview</b>	<b>2</b>
2.1 Kullanılan Teknolojiler	2
2.2 Dağıtım Ortamı	2
<b>3. Deployment Process</b>	<b>3</b>
3.1 Kurulum Adımları	3
3.2 Demo Hazırlığı İçin Ekstra Notlar	3
<b>4. Configuration Plan</b>	<b>4</b>
4.1 Uygulama Ayarları	4
4.2 Veritabanı Yapısı (SQLite)	4
4.3 Dosya ve Dizin Yapısı	4

# 1. Giriş

Bu dağıtım planı, İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi'nin demo sunumu öncesinde nasıl kurulup çalıştırıldığını açıklamaktadır. Plan, uygulamanın kurulum ortamı, kullanılan araçlar, dağıtım süreci ve yapılandırma detaylarını kapsamaktadır.

---

## 2. Deployment Overview

### 2.1 Kullanılan Teknolojiler

- **Python 3.10+**: Backend geliştirme dili
- **Flask**: Web sunucusu ve arayüz servisi
- **SQLite**: Hafif veritabanı yönetim sistemi
- **BeautifulSoup, Feedparser**: RSS veri çekimi için
- **NLTK, Scikit-learn**: Doğal dil işleme ve LDA modeli için
- **CSS + HTML**: Web arayüzü için

### 2.2 Dağıtım Ortamı

- **Çalışma Sistemi**: Windows 10 / Ubuntu 22.04
  - **Kurulum Yeri**: Yerel makine
  - **Çalışma Ortamı**: VSCode + Terminal / Google Colab
  - **Tarayıcı**: Google Chrome veya Firefox
-

## 3. Deployment Process

### 3.1 Kurulum Adımları

**1. Kod deposunun klonlanması:**

```
git clone https://github.com/yourteam/ai-climate-action.git  
cd AI-Based-Action-Recommender-for-Climate-Crisis
```

**2. Gerekli bağımlılıkların kurulması:**

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

**3. Veritabanının oluşturulması:**

```
python modules/db_helper.py
```

**4. RSS veri çekme işlemi:**

```
python modules/rss_scraper.py
```

**5. NLP işlemleri ve kategorilendirme:**

```
python modules/text_cleaner.py  
python modules/lda_categorizer.py
```

**6. Uygulamanın başlatılması:**

```
python app.py
```

**7. Arayüze erişim:**

Tarayıcıdan şu adres ziyaret edilir: <http://127.0.0.1:5000>

Manuel Veri eklemek için tarayıcıdan şu adres ziyaret edilir: <http://localhost:5000/admin>

### 3.2 Demo Hazırlığı İçin Ekstra Notlar

- Manuel eklenecek haberler öncesinde sunum için hazır hale getirilmelidir.
  - Her modül, demo sırasında bağımsız şekilde test edilebilecek biçimde çalıştırılabilir halde olmalıdır.
  - Flask debug modu aktif bırakılmalıdır (debug=True).
-

## 4. Configuration Plan

### 4.1 Uygulama Ayarları

app.py içinde port ve debug seçenekleri:

```
-app.run(debug=True, port=5000)
```

### 4.2 Veritabanı Yapısı (SQLite)

- Dosya: news.db
- Tablo: news
- Alanlar: id, title, content, category, suggestion

### 4.3 Dosya ve Dizin Yapısı

```
|— app.py
|— config.py
|— cleanup_duplicates.py (helper module)
|— static/
|   |— styles.css
|— templates/
|   |— index.html
|   |— admin.html
|— modules/
|   |— db_helper.py
|   |— rss_scraper.py
|   |— text_cleaner.py
|   |— lda_categorizer.py
|   |— manuel_entry.py
|   |— language_filter.py
|   |— structures.py
|— news.db
|— requirements.txt
|— README.md
```