



## Katkıda Bulunanların Listesi

- Ali Hakan Kıncal
- Barış Emre Ahi

## Task Matrix

Görev	Atanan Kişi
RSS Veri Çekme Testi	Ali Hakan Kıncal
Metin Temizleme Testi	Barış Emre Ahi
Makine Öğrenmesi Kategorilendirme Testi	Mehmet Begun
Öneri Motoru Testi	Ahmet Berkay Avan
Veritabanı Testi	Ahmet Berkay Avan
Arayüz Testi	Ali Hakan Kıncal, Barış Emre Ahi
Performans Testi	Mehmet Begun
Güvenlik Testi (SQL Injection, XSS, vb.)	Barış Emre Ahi
Genel Kod İnceleme	Ahmet Berkay Avan, Ali Hakan Kıncal, Barış Emre Ahi, Mehmet Begun

Katkıda Bulunanların Listesi	1
<b>İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi - Quality Assurance (QA) Planı</b>	<b>2</b>
1. Giriş	2
1.1 QA Planının Amacı	2
1.2 QA Planının Kapsamı	2
2. Test Stratejisi	3
2.1 Test Türleri (Types of Testing)	3
2.2 Test Case Örnekleri	4
3. Kalite Metrikleri	4
4. Test Planı	5
5. Hata Yönetimi ve Takip Süreci	6
5.1 Hata Önceliklendirme Seviyeleri	6
5.2 Hata Çözüm Süreci	6
6. Sonuç ve Gelecek Planlar	6

# İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi - Quality Assurance (QA) Planı

---

## 1. Giriş

Bu QA Planı, İklim Krizi İçin Yapay Zeka Destekli Eylem Öneri Sistemi'nin hata oranını minimize etmek, sistemin güvenilirliğini artırmak ve kullanıcı deneyimini iyileştirmek amacıyla uygulanacak test süreçlerini tanımlar.

### 1.1 QA Planının Amacı

- Projenin güvenilir, hatasız ve verimli bir şekilde çalışmasını sağlamak
- Hata oranlarını azaltmak ve yazılımın performansını optimize etmek
- Otomatik ve manuel test süreçlerini planlamak ve yürütmek

### 1.2 QA Planının Kapsamı

Bu QA planı aşağıdaki sistem bileşenlerinin test edilmesini kapsar:

- ✓ **Veri toplama modülü** → RSS kaynaklarından veri çekme doğruluğu
  - ✓ **NLP analizi ve öneri motoru** → Anahtar kelime tespiti ve öneri doğruluğu
  - ✓ **Veritabanı yönetimi** → Verinin doğru saklanıp saklanmadığı
  - ✓ **Kullanıcı arayüzü** → Web arayüzü ve kullanıcı deneyimi testleri
-

## 2. Test Stratejisi

Proje için hem manuel hem de otomatik test yöntemleri uygulanacaktır.

### 2.1 Test Türleri (Types of Testing)

Test Türü	Test Kapsamı
Birim Testleri (Unit Tests)	RSS çekme, metin temizleme, makine öğrenmesi ve veritabanı işlemleri test edilir.
Entegrasyon Testleri	Veri çekme, veritabanı ve makine öğrenmesi modülleri birlikte test edilir.
Fonksiyonel Testler	Kullanıcının haberleri görüntüleyip önerileri alabilmesi test edilir.
Performans Testleri	Büyük veri kümeleriyle sistemin hız ve verimliliği ölçülür.
Güvenlik Testleri	SQL Injection, XSS gibi güvenlik açıkları test edilir.
Kullanıcı Kabul Testleri (UAT)	Kullanıcıların arayüzü kullanarak sistemin doğru çalıştığını doğrulaması sağlanır.

## 2.2 Test Case Örnekleri

- **TC-01:** RSS Kaynaklarından Veri Çekme

**Testin Amacı:** Sistem, belirlenen RSS kaynaklarından haberleri çekmelidir.

**Ön Koşullar:** RSS bağlantıları geçerli olmalıdır.

**Test Adımları:** 1. `fetch_rss()` fonksiyonunu çalıştır  
2. RSS verilerinin veritabanına kaydedildiğini doğrula

**Beklenen Sonuç:** Haberler başarıyla çekilmeli ve `news.db`'ye kaydedilmelidir

**Durum:** Başarılı

- **TC-02:** Kategorize Edilen Haberlerin Doğruluğu

**Testin Amacı:** LDA modeliyle haberlerin doğru kategoriye ayrıldığını kontrol eder.

**Ön Koşullar:** Veritabanında en az 10 haber olmalıdır.

**Test Adımları:** 1. `categorize_news()` fonksiyonunu çalıştır  
2. Haberlerin `category` alanlarının boş olmadığını kontrol et

**Beklenen Sonuç:** Tüm haberler uygun bir kategoriye atanmış olmalıdır.

**Durum:** Başarılı

## 3. Kalite Metrikleri

Sistemin kalitesini ölçmek için aşağıdaki **metrikler kullanılacaktır**:

Metrik	Açıklama	Ölçüm Yöntemi
Kod Kapsamı (Code Coverage)	Test edilen kodun yüzdesi	<code>pytest --cov</code> gibi araçlarla ölçülür
Yanıt Süresi (Response Time)	API isteklerinin yanıt süresi	Postman veya JMeter ile ölçülür
Hata Oranı (Bug Rate)	Her sürümde tespit edilen hata oranı	JIRA veya GitHub Issues kullanılır
Veritabanı Yanıt Süresi	SQL sorgularının hızını ölçer	<code>EXPLAIN ANALYZE</code> ile ölçülür
Kullanıcı Memnuniyeti	Kullanıcı geri bildirim puanı	Anketler ve test kullanıcıları

---

## 4. Test Planı

Test Adımı	Açıklama	Beklenen Sonuç	Durum
<b>RSS verisi çekme testi</b>	Haber kaynaklarından RSS verisi çekiliyor mu?	Haber veritabanına başarılı ekleme	Başarılı
<b>Metin Temizleme Testi</b>	HTML etiketleri ve gereksiz karakterler temizleniyor mu?	Temizlenmiş ve anlamlı bir haber özeti elde edilmeli	Başarılı
<b>Makine Öğrenmesi Kategorilendirme Testi</b>	LDA modeli haberleri doğru kategorize ediyor mu?	Her haber bir kategoriye atanmalı.	Başarılı
<b>Öneri Motoru Testi</b>	NLP analizi sonrası doğru öneriler üretiliyor mu?	Kullanıcılara ilgili kategoriye uygun öneriler sunulmalı	Başarılı
<b>Veritabanı Testi</b>	Haber veritabanı doğru çalışıyor mu?	Haberler ve kategorileri doğru saklanmalı	Başarılı
<b>Performans Testi</b>	RSS verisi büyük miktarda çekildiğinde sistem nasıl çalışıyor?	1000+ haber işlendiğinde sistem çökmeden çalışmalı	Devam Ediyor

---

## 5. Hata Yönetimi ve Takip Süreci

Hata yönetimi, **JIRA**, **Trello** veya **GitHub Issues** üzerinden yapılacaktır.

### 5.1 Hata Önceliklendirme Seviyeleri

Öncelik Seviyesi	Tanım
<b>Kritik (Critical)</b>	Sistemin tamamen çalışmasını engelleyen hatalar.
<b>Yüksek (High)</b>	Kullanıcının temel işlevleri yerine getirmesini engelleyen hatalar.
<b>Orta (Medium)</b>	Kullanıcı deneyimini olumsuz etkileyen ancak işleyişi bozmayan hatalar.
<b>Düşük (Low)</b>	Kullanıcı deneyimini etkileyen küçük sorunlar veya önerilen geliştirmeler.

### 5.2 Hata Çözüm Süreci

- 1- Hata tespit edilir ve JIRA/Trello/GitHub Issues'a kaydedilir.
- 2- Hata önceliği: kritik, yüksek, orta, düşük.
- 3- Geliştirici ekibi hatayı çözer ve kodu tekrar test eder.
- 4- QA ekibi düzeltmeyi kontrol eder, hata giderilmişse kapatılır.

## 6. Sonuç ve Gelecek Planlar

#### ✓ Mevcut Sorunlar:

- Otomatik testlerin artırılması gerekiyor.
- Performans testleri için daha fazla stres testi yapılmalı.

#### ✓ Gelecekteki Geliştirmeler:

- Test süreçlerini CI/CD pipeline içine entegre etmek.
- Daha kapsamlı kullanıcı deneyimi testleri yapmak.
- Sistemin mobil uyumluluğunu test etmek.