

-----

-----

## # Akıllı İçerik Platformu (FrameFlow AI) - Teknik Arşiv ve Kullanıcı Kılavuzu

**\*\*Versiyon:\*\*** 1.0 (Canlı - GCS Entegrasyonlu)  
**\*\*Oluşturan:\*\*** b!g & Gemini  
**\*\*Canlı URL:\*\*** [https://akilli-icerik-platformu.onrender.com/`](https://akilli-icerik-platformu.onrender.com/)

-----

### ## BÖLÜM 1: Basit Kullanıcı Rehberi (Teknik Olmayan Açıklama)

#### ### Projeniz Ne İşe Yarıyor? (Basit Dil)

Bu platform, dijital bir **\*\*Akıllı Öğrenme Asistanıdır\*\***.

Uzun bir ders videosu, saatlerce süren bir ses kaydı veya onlarca sayfalık bir PDF belgesi düşünün. Bu

Bu platform, sizin "zaman alan" tüm bu içerikleri (ses, video, PDF, ders notu) yüklediğiniz bir sistem

**\*\*Özetle:\*\*** Siz bir saatlik ders videosunu yüklersiniz, o size 2 dakika içinde dersin özetini, anahtar

#### ### Kime Fayda Sağlar?

- \* **\*\*Öğrenciler:\*\*** Sınavlara hazırlanırken saatlerce video izlemek yerine, 5 dakikada dersin özetini
- \* **\*\*Profesyoneller ve Yetişkinler:\*\*** Yoğun iş temposunda kaçırdıkları bir toplantının ses kaydını ve

#### ### Bu Proje Neden Farklı?

Çünkü bu platform, sizin adınıza **\*\*kalıcı bir hafıza\*\*** oluşturur. Oluşturulan her rapor, size özel bir

-----

### ## BÖLÜM 2: Son Kullanıcı Kılavuzu (Platform Nasıl Kullanılır?)

Akıllı İçerik Platformu'nu kullanmak için bir "Erişim Kodu"na (Token) ihtiyacınız vardır. Bu kod, sizi

#### ### 1\ Adım: Erişim Kodu (Token) Nasıl Alınır?

Platformumuzda yeni bir hesap oluşturmak ve Token almak ücretsiz ve anlıktır:

1. Canlı platform adresine gidin: **\*\*`https://akilli-icerik-platformu.onrender.com/`\*\***
2. Ana sayfada **\*\*" Erişim Kodu (Token)"\*\*** alanının altında, **\*\*"Yeni Kullanıcı Kaydı"\*\*** bölümünü bul
3. Formu doldurun:
  - \* **\*\*Kullanıcı ID:\*\*** (Örn: `ali\_yilmaz\_123`)
  - \* **\*\*E-posta:\*\*** (Geçerli bir e-posta)
  - \* **\*\*Şifre:\*\*** (Güvenli bir şifre)
4. **\*\*"Kayıt Ol"\*\*** butonuna tıklayın.
5. Sistem size anında yeni bir **\*\*Token\*\*** (Erişim Kodu) verecek ve bu kodu otomatik olarak tarayıcınıza

#### ### 2\ Adım: Bir İçerik Nasıl Analiz Edilir?

Token'ınızı aldıktan sonra, analiz işlemi çok basittir:

1. Giriş yaptığınızda, **\*\*" İçerik Yükle & Analiz Et"\*\*** bölümü otomatik olarak görünür.
2. **\*\*Dosya Yükleme:\*\*** "Dosya Seç" butonuna tıklayarak bilgisayarınızdan `.mp3`, `.pdf`, `.docx` veya
3. **\*\*VEYA YouTube Linki:\*\*** YouTube URL'si alanına, analiz etmek istediğiniz videonun linkini yapıştır
4. **\*\*"Analizi Başlat"\*\*** butonuna tıklayın.
5. İlerleme çubuğu (`progress-bar`) dolana kadar bekleyin. Analiz süresi, içeriğin uzunluğuna bağlı o

#### ### 3\ Adım: Raporuma Nasıl Erişirim?

Analiz bittiğinde, raporunuz doğrudan ekranınızda görünecektir:

- \* **\*\*" Analiz Raporu"\*\*** bölümü ekranda belirir.
- \* Raporun 8 başlığını (Özet, Sözlük, Quiz vb.) doğrudan sayfada okuyabilirsiniz.

\* Raporun **\*\*kalıcı kopyasını\*\*** indirmek veya görüntülemek için, rapor başlığının hemen altındaki **\*\***

-----

## ## BÖLÜM 3: Detaylı Teknik Dokümantasyon (Sistem Mimarisi)

Bu bölümde, projenin teknik iskeleti, kullanılan teknolojiler ve kodlama mantığı detaylandırılmıştır.

### ### 1\. Sistem Mimarisi (Genel Bakış)

Bileşen	Teknoloji	Amaç
Backend (API)	FastAPI (Python)	Tüm mantığın işlediği, API isteklerini kabul eden sunucu.
Sunucu (Server)	Uvicorn   FastAPI'yi çalıştıran ASGI sunucusu.	
Frontend (UI)	HTML5, CSS3, JavaScript	Kullanıcının etkileşime girdiği arayüz.
Yapay Zeka (AI)	OpenAI (GPT-4o, Whisper)	İçerik analizi, raporlama ve ses/video metne dökme
Canlı Yayın (Deploy)	Render (PaaS)	Projenin internette canlı yayınlanmasını sağlayan bulut p
Kalıcı Depolama	Google Cloud Storage (GCS)	Oluşturulan tüm raporların kalıcı ve güvenli olar
Sürüm Kontrolü	Git & GitHub	Kod yönetimi ve Render'a otomatik dağıtım (CI/CD).

### ### 2\. Backend Mimarisi (`backend/main.py`)

Backend, iki ana API uç noktası üzerine kuruludur:

#### #### A. `/register` (POST)

- \*\*İşlevi:\*\*** Yeni kullanıcı kaydı yapar ve dinamik Token oluşturur.
- \*\*Çalışma Mantığı:\*\***
  - `UserRegistration` Pydantic modelini kullanarak `user\_id`, `email` ve `password` alır.
  - `secrets.token\_urlsafe(32)` kullanarak kriptografik olarak güvenli bir Token oluşturur.
  - Bu Token'ı ve kullanıcı bilgilerini **\*\*bellekteki\*\*** `USERS\_DB` sözlüğüne (dictionary) kaydeder.
  - Kullanıcıya Token'ı ve başarı mesajını JSON olarak döndürür.
- \*\*Not:\*\*** Render'ın geçici dosya sistemi nedeniyle, `users.json` dosyası kalıcı değildir. `USERS\_DB`

#### #### B. `/analiz-et` (POST)

- \*\*İşlevi:\*\*** Ana analiz ve raporlama iş yükünü yönetir.
- \*\*Çalışma Mantığı (5 Adımda):\*\***
  - \*\*Güvenlik (Token Kontrolü):\*\*** `X-API-TOKEN` başlığını alır. `get\_user\_id` fonksiyonu ile `USE`
  - \*\*İçerik Okuma:\*\*** Yüklenen `dosya` veya `youtube\_url`'yi kontrol eder.
    - `validate\_file` ile dosya boyutu ve uzantısı kontrol edilir.
    - İlgili okuyucu fonksiyonu çağrılır:
      - `read\_audio` (Whisper API)
      - `read\_pdf` (PyPDF2)
      - `read\_docx` (python-docx)
      - `read\_pptx` (python-pptx)
      - `read\_image` (GPT-4o Vision API)
      - `download\_youtube\_audio` (pytube + Whisper API)
  - \*\*Yapay Zeka Raporlaması:\*\*** Okunan metin, `RAPOR\_PROMPTU` (8 başlık talimatını içeren) ile bir
  - \*\*Kalıcı Depolama (GCS):\*\***
    - Render'ın çevresel değişkenlerinden `GCS\_SA\_KEY` (Google Cloud JSON Anahtarı) okunur.
    - `storage.Client.from\_service\_account\_info()` ile GCS istemcisi başlatılır.
    - `GCS\_BUCKET\_NAME` (`akilli-icerik-raporlari-bbkgzn`) seçilir.
    - Dosya yolu oluşturulur: `f"{user\_id}/{temiz\_ad}\_{zaman\_damgasi}.md"`
    - `blob.upload\_from\_string()` metodu çağrılır.
    - \*\*Kritik Düzeltme (UTF-8):\*\*** Türkçe karakter sorununu çözmek için `data=rapor\_metni.encode`
  - \*\*Yanıt Dönüşü:\*\*** Hem `rapor\_markdown` metni hem de kalıcı `dosya\_url`'si (örn: `https://stora`)

### ### 3\. Frontend Mimarisi (`frontend/`)

- \*\*`index.html`:\*\*** Arayüzün iskeletidir. `auth-section` (Kayıt/Token) ve `analysis-section` (Yüklem
- \*\*`script.js`:\*\*** Frontend'in beynidir.
  - \*\*Token Yönetimi:\*\*** `localStorage.setItem(TOKEN\_STORAGE\_KEY, token)` kullanarak Token'ı tarayı
  - \*\*Kayıt Akışı:\*\*** `registerForm` olayını dinler, `/register` endpoint'ine `fetch` ile POST iste
  - \*\*Analiz Akışı:\*\*** `startAnalysis` fonksiyonu, `FormData` oluşturur, seçilen dosyayı veya URL'y
  - \*\*Rapor Gösterimi:\*\*** Başarılı yanıt aldığında, `marked.parse(data.rapor\_markdown)` kullanarak l

### ### 4\. Dağıtım ve DevOps (Render & Güvenlik)

```
* **`Procfile`:** `web: uvicorn backend.main:app --host 0.0.0.0 --port $PORT` komutunu içerir. Rende
* **`.gitignore`:** `*.env`, `users.json`, `reports/` gibi hassas ve gereksiz dosyaların GitHub'a yü
* **`LICENSE`:** CC BY-NC 4.0 lisansı ile kodun ticari kullanımı kısıtlanmıştır.
* **Render Çevresel Değişkenleri:**
  * `OPENAI_API_KEY`: GPT ve Whisper'a erişim için.
  * `GCS_SA_KEY`: Google Cloud Storage'a yazma erişimi için (İndirilen JSON dosyasının tam metnini
```

-----

-----

## ## BÖLÜM 4: Projenin Tam Kod Arşivi (Final Versiyon)

Bu bölümde, projenizin canlıda çalışan tüm dosyalarının eksiksiz kodları yer almaktadır.

### 1\. `backend/main.py` (Ana Sunucu Kodu - UTF-8 Düzeltmeli)

```
```python
```

```
# (Tam kod burada - yukarıdaki içerikte yer aldığı gibi)
```

```
```
```

... (İzleyen bölümlerin tam içeriği yukarıdaki metinde yer almaktadır) ...