🚀 CV Detector Backend - Hızlı Başlangıç

Adım Adım Kurulum

1️⃣ Sistem Gereksinimlerini Kurun

Tesseract OCR

Ubuntu/Debian:

```bash
sudo apt-get update
sudo apt-get install tesseract-ocr tesseract-ocr-tur tesseract-ocr-eng
sudo apt-get install poppler-utils  # PDF işleme için
```

macOS:

```bash
brew install tesseract tesseract-lang poppler
```

Windows:

1. Tesseract Installer indirin
2. Kurulum sırasında "Additional language data" seçeneğinden Turkish ve English seçin
3. Poppler for Windows indirin ve PATH'e ekleyin

2️⃣ Projeyi Klonlayın

```bash
git clone <repository-url>
cd Find_a_Job_with_CV_Detector/backend
```

3️⃣ Python Sanal Ortamı Oluşturun

```bash
```

```
# Sanal ortam oluştur
python3 -m venv venv

# Aktif et (Linux/macOS)
source venv/bin/activate

# Aktif et (Windows)
venv\Scripts\activate
```

4️⃣ Bağımlılıkları Yükleyin

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

5️⃣ Gerekli Klasörleri Oluşturun

```bash
mkdir -p storage/resumes
```

6️⃣ init.py Dosyalarını Oluşturun

Her klasörde `__init__.py` dosyası olmalı:

```bash
touch app/__init__.py
touch app/api/__init__.py
touch app/config/__init__.py
touch app/infra/__init__.py
touch app/services/__init__.py
```

7️⃣ Server'ı Başlatın

```bash
uvicorn app.main:app --reload --host 0.0.0.0 --port 8000
```

✅ Server çalışıyorsa şu adreslere erişebilirsiniz:

- API: http://localhost:8000
- Swagger UI: http://localhost:8000/docs
- Health Check: http://localhost:8000/api/health

🧪 İlk Test

1. Health Check

```bash
curl http://localhost:8000/api/health
```

Beklenen çıktı:

```json
{"status":"ok"}
```

2. CV Yükle ve Test Et

Örnek bir CV dosyası oluşturun veya hazır bir CV kullanın:

```bash
# CV yükle
curl -X POST "http://localhost:8000/api/upload" \
  -F "file=@your_cv.pdf"
```

Dönen response'dan `analysis_id`'yi alın:

```json
{
  "analysis_id": "abc-123-def-456",
  "status": "done"
}
```

3. Sonuçları Kontrol Et

```bash
curl http://localhost:8000/api/results/abc-123-def-456
```

🌐 Frontend ile Entegrasyon

Backend çalıştıktan sonra, frontend'inizin JavaScript kodunda API URL'ini güncelleyin:

```javascript
```

```
// Frontend'te API base URL
const API_BASE_URL = "http://localhost:8000/api";

// CV upload
const formData = new FormData();
formData.append('file', fileInput.files[0]);

const response = await fetch(`${API_BASE_URL}/upload`, {
    method: 'POST',
    body: formData
});

const result = await response.json();
console.log(result.analysis_id);

// Sonuçları al
const resultsResponse = await fetch(
    `${API_BASE_URL}/results/${result.analysis_id}`
);
const analysisResults = await resultsResponse.json();
console.log(analysisResults);
```

📊 Swagger UI ile Test

1. Tarayıcıda http://localhost:8000/docs adresini açın
2. `POST /api/upload` endpoint'ini genişletin
3. "Try it out" butonuna tıklayın
4. Bir dosya seçin ve "Execute" yapın
5. Dönen `analysis_id`'yi kopyalayın
6. `GET /api/results/{analysis_id}` endpoint'ini test edin

🐛 Sık Karşılaşılan Sorunlar

Sorun: "Tesseract not found"

Çözüm:

```
bash

# Tesseract path'ini kontrol edin
which tesseract  # Linux/macOS
where tesseract  # Windows
```

Eğer bulunamıyorsa, `text_extraction_service.py` içinde path belirtin:

```python
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r'/usr/local/bin/tesseract'
```

Sorun: "No module named 'fitz'"

Çözüm:

```bash
pip install PyMuPDF
```

Sorun: PDF'den metin çıkaramıyor

Çözüm: Poppler kurulu olmalı:

```bash
# Ubuntu
sudo apt-get install poppler-utils

# macOS
brew install poppler
```

Sorun: CORS hatası

Çözüm: `app/main.py` içinde CORS ayarlarını kontrol edin. Frontend URL'iniz varsa ekleyin:

```python
app.add_middleware(
    CORSMiddleware,
    allow_origins=["http://localhost:5500", "http://127.0.0.1:5500"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

🔍 Logları İzleme

Server loglarında işlemleri takip edebilirsiniz:

```
[ProcessingService] Text extraction başlıyor: /path/to/file.pdf
[ProcessingService] 1234 karakter metin çıkarıldı
[ProcessingService] Temel bilgiler: 1 email, 1 telefon
[ProcessingService] 8 teknoloji tespit edildi
[ProcessingService] İşlem tamamlandı: abc-123-def-456
```

📈 Performans İpuçları

- **Küçük dosyalar**: ~1-2 saniye
- **Orta dosyalar (1-2 sayfa PDF)**: ~3-5 saniye
- **Büyük dosyalar veya OCR gereken**: ~10-30 saniye

OCR işlemleri CPU yoğundur. Production'da asenkron işleme (Celery) kullanın.

🎉 Başarıyla Kuruldu!

Artık backend'iniz çalışıyor ve frontend ile entegre edebilirsiniz.

Sonraki Adımlar:

1. ✅ Frontend'inizi backend'e bağlayın
2. ✅ Teknoloji sözlüğünü özelleştirin (`app/config/tech_dictionary.json`)
3. ✅ Daha fazla test CV'si ile deneyin
4. ✅ Production için ayarları yapın

📞 Yardım

Sorun yaşıyorsanız:

1. Terminal loglarını kontrol edin
2. `http://localhost:8000/docs` adresinden Swagger UI'a bakın
3. GitHub Issues'a sorun açın

Kolay gelsin! 🚀