

FAZ 1: BACKEND VE AI MOTORU

1. Altyapı Kurulumu

- [] Proje ana dizininde /backend klasörünü oluştur.
- [] Terminalde `python -m venv venv` komutuyla sanal ortam oluştur.
- [] Sanal ortamı aktif et (Windows: `venv\Scripts\activate`, Mac/Linux: `source venv/bin/activate`).
- [] `requirements.txt` dosyasını oluştur ve şu paketleri ekle:
 - `fastapi`
 - `uvicorn`
 - `python-multipart`
 - `pandas`
 - `openpyxl`
 - `google-generativeai`
 - `python-dotenv`
- [] Paketleri kur: `pip install -r requirements.txt`
- [] `.env` dosyası oluştur ve içine `GOOGLE_API_KEY=senin_api_anahtarin` satırını ekle.

2. Sunucu Temeli (main.py)

- [] `backend/main.py` dosyasını oluştur.
- [] FastAPI uygulamasını (`app = FastAPI()`) başlat.
- [] `CORSMiddleware` ekleyerek `http://localhost:5173` adresine (Frontend) izin ver.
- [] Pydantic çıktı modelini tanımla: `class LeadAnalysis(BaseModel): score: int, reason: str, action: str, customer_name: str.`

3. Gemini Konfigürasyonu

- [] `google.generativeai` kütüphanesini import et ve API anahtarını `.env`den çekerek konfigüre et.
- [] Modeli tanımlarken `generation_config={"response_mime_type": "application/json"}` parametresini ekle (Bu çok kritik).

4. Analiz Endpoint'i (/analyze)

- [] `POST /analyze` endpoint'ini yaz. Parametre olarak `file: UploadFile` almalı.
- [] Dosyayı `pandas` ile oku (`pd.read_excel` veya `pd.read_csv`).
- [] Veriyi JSON formatına çevir.
- [] Gemini'ye şu talimatla gönder: "Bu listeyi analiz et. Her satır için skor (0-100), sebep, aksiyon ve müşteri adını içeren bir JSON listesi dön."
- [] Gelen yanıtı return et.

5. Test

- [] Sunucuyu başlat: `uvicorn main:app --reload`
- [] Tarayıcıda `http://localhost:8000/docs` adresine git.
- [] Örnek bir Excel dosyası yükleyip "Execute" diyerek JSON yanıtı aldığını doğrula.