

Elbette. İşte tam olarak bu isteği yerine getirecek, Replit'teki AI kodlama asistanınıza doğrudan yapıştırabileceğiniz, son derece detaylı bir master proje prompt'u. Bu prompt, AI'a sadece ne yapacağını değil, aynı zamanda hangi dosyayı değiştireceğini, hangi kodları yazacağını ve Azure'dan ElevenLabs kalitesinde ses almak için hangi akıllı mantığı kuracağını adım adım talimatlarla anlatır. MASTER PROMPT: AZURE TTS KALİTE YÜKSELTME

ENTTEGRASYON UĞÖREV: Mevcut AI sesli asistan projesindeki ElevenLabs TTS servisini, en yüksek kalitede ve doğal Türkçe ses üretecek şekilde Microsoft Azure Neural TTS ile değiştir. Bu entegrasyon, düz metni akıllıca analiz edip, sese duygu, vurgu ve doğallık katmak için dinamik olarak SSML (Speech Synthesis Markup Language) üreten bir mantık içermelidir. TEKNİK PLAN: Hedef Dosya: main.py Gereklili Kütüphane:

azure-cognitiveservices-speech Güvenlik: Azure API anahtarları Replit Secrets'tan okunacak. Çekirdek Mantık: Anthropic'ten gelen düz metni, SSML'e dönüştüren yeni bir fonksiyon oluşturulacak. ADIM 1: GEREKLİ KÜTÜPHANENİN KURULUMU requirements.txt dosyasına aşağıdaki satırı ekle: code Codedownloadcontent_copyexpand_less azure-cognitiveservices-speech

ADIM 2: BACKEND KODUNU YENİDEN YAPILANDIR (main.py) main.py dosyasında aşağıdaki değişiklikleri yap: A. API Anahtarlarını Yükle: Dosyanın başına, diğer anahtarların yanına Azure anahtarlarını da ekle. code Pythondownloadcontent_copyexpand_less IGNORE_WHEN_COPYING_START IGNORE_WHE N_COPYING_END # main.py

```
# ... diğer importlar ...  
import azure.cognitiveservices.speech as speechsdk
```

```
# ...
```

```
# --- API ANAHTARLARINI GÜVENLİ YERDEN ÇEK ---
```

```
# ... diğer anahtarlar ...
```

```
AZURE_SPEECH_KEY = os.environ.get('AZURE_SPEECH_KEY')
```

```
AZURE_SPEECH_REGION = os.environ.get('AZURE_SPEECH_REGION')
```

B. elevenlabs_text_to_speech Fonksiyonunu Değiştir: Mevcut elevenlabs_text_to_speech fonksiyonunu tamamen sil ve yerine aşağıdaki iki yeni fonksiyonu ekle. ADIM 3: AKILLI SSML ÜRETİCİ FONKSİYONU (create_ssml_for_text) Bu yeni fonksiyon, AI'dan gelen düz metni analiz edip Azure'un anlayacağı zengin SSML formatına dönüştürecek. Bu, kaliteyi artıran sihirli adımdır. code

```
Pythondownloadcontent_copyexpand_less IGNORE_WHEN_COPYING_START IGNORE_WHE N_COPYING_END # main.py'ye eklenecek YENİ FONKSİYON 1
```

```
def create_ssml_for_text(text, voice_name="tr-TR-EmelNeural"):
```

```
    """
```

```
    Anthropic'ten gelen düz metni analiz eder ve Azure için dinamik SSML oluşturur.
```

```
    Bu fonksiyon, sese duygu, vurgu ve doğallık katar.
```

```
    """
```

```

# Kural 1: Neşeli karşılama ve tebrikler
positive_words = ["merhaba", "hoş geldin", "harika", "mükemmel", "tebrikler"]
if any(word in text.lower() for word in positive_words):
    # Neşeli bir tonla söylet
    ssml_body = f'<mstts:express-as style="cheerful">{text}</mstts:express-as>'

# Kural 2: Soruları daha doğal hale getirme
elif text.endswith('?'):
    # Sorunun son kelimesinin perdesini hafifçe yükselterek doğal bir soru tonu ver
    words = text.split()
    question_part = " ".join(words[-3:]) # Son 3 kelimeyi al
    main_part = " ".join(words[:-3])
    ssml_body = f'{main_part} <prosody pitch="+15%">{question_part}</prosody>'

# Kural 3: Vurgu ekleme (Örnek: tırnak içindeki kelimeler)
elif '"' in text:
    # Tırnak içindeki kelimeleri daha vurgulu yap
    parts = text.split('"')
    ssml_body = f'{parts[0]}<emphasis level="strong">{parts[1]}</emphasis>{parts[2]}'

else:
    # Varsayılan durum: Kural yoksa düz metin
    ssml_body = text

# Final SSML'i oluştur
ssml_string = f'"""
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xmlns:mstts="http://www.w3.org/2001/mstts" xml:lang="tr-TR">
    <voice name="{voice_name}">
        {ssml_body}
    </voice>
</speak>
"""

return ssml_string

```

ADIM 4: YENİ AZURE TTS FONKSİYONU (azure_text_to_speech) Bu fonksiyon, yukarıda oluşturulan SSML'i alıp Azure'a gönderecek ve yüksek kaliteli sesi üretecek. code

```

Pythondownloadcontent_copyexpand_lessIGNORE_WHEN_COPYING_STARTIGNORE_WHE
N_COPYING_END # main.py'ye eklenecek YENİ FONKSİYON 2

```

```

def azure_text_to_speech(ssml_string):
    """
    Verilen SSML metnini kullanarak Azure'dan yüksek kaliteli ses sentezler.
    """

```

```
speech_config = speechsdk.SpeechConfig(subscription=AZURE_SPEECH_KEY,
region=AZURE_SPEECH_REGION)
```

```
# En yüksek kalite için çıkış formatını ayarla
```

```
speech_config.set_speech_synthesis_output_format(speechsdk.SpeechSynthesisOutputFormat.Audio48Khz192KbitrateMonoMp3)
```

```
# Bellekteki bir akışa sentezleme yapmak için
```

```
pull_stream = speechsdk.audio.PullAudioOutputStream()
```

```
stream_config = speechsdk.audio.AudioOutputConfig(stream=pull_stream)
```

```
speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config,
audio_config=stream_config)
```

```
# SSML'i kullanarak sesi sentezle
```

```
result = speech_synthesizer.speak_ssml_async(ssml_string).get()
```

```
if result.reason == speechsdk.ResultReason.SynthesizingAudioCompleted:
```

```
    print("Azure TTS başarılı.")
```

```
    return result.audio_data # Geriye ham ses verisini (bytes) döndür
```

```
elif result.reason == speechsdk.ResultReason.Canceled:
```

```
    cancellation_details = result.cancellation_details
```

```
    print(f"Azure TTS iptal edildi: {cancellation_details.reason}")
```

```
    if cancellation_details.reason == speechsdk.CancellationReason.Error:
```

```
        print(f"Hata detayı: {cancellation_details.error_details}")
```

```
    return None
```

ADIM 5: ANA İŞ AKIŞINI GÜNCELLE (process_audio_route) Son olarak, ana API endpoint'inde ElevenLabs çağrısını bu yeni, iki adımlı Azure süreciyle değiştir. code
Pythondownloadcontent_copyexpand_lessIGNORE_WHEN_COPYING_STARTIGNORE_WHEN_COPYING_END # main.py içindeki bu fonksiyonu GÜNCELLE

```
@app.route('/process-audio', methods=['POST'])
```

```
def process_audio_route():
```

```
    try:
```

```
        # ... (Clerk doğrulama, Azure STT, Neon DB işlemleri aynı kalır) ...
```

```
        # user_id = ...
```

```
        # user_text = ...
```

```
        # history = ...
```

```
        ai_text_response = anthropic_get_response(user_text, history)
```

```
        # save_history_to_db(...)
```

```
        # --- ESKİ KOD (SİLİNECEK) ---
```

```
        # ai_audio_response = elevenlabs_text_to_speech(ai_text_response)
```

```
# --- YENİ, YÜKSEK KALİTELİ AZURE AKIŞI ---
```

```
# 1. Akıllı SSML oluştur
```

```
ssml_to_speak = create_ssml_for_text(ai_text_response)
```

```
# 2. SSML'i kullanarak sesi Azure'da sentezle
```

```
ai_audio_response = azure_text_to_speech(ssml_to_speak)
```

```
if ai_audio_response:
```

```
    return ai_audio_response, 200, {'Content-Type': 'audio/mpeg'}
```

```
else:
```

```
    return "Ses üretilemedi", 500
```

```
except Exception as e:
```

```
    print(f"Bir hata oluştu: {e}")
```

```
    return jsonify({"error": str(e)}), 500
```

Bu prompt'u uyguladığınızda, projeniz artık ElevenLabs'e ihtiyaç duymadan, Azure'un gücünü kullanarak dinamik, duygusal ve yüksek kaliteli Türkçe sesler üretebilen profesyonel bir altyapıya sahip olacaktır.