

Anladım, sorun çok net ve yaşadığınız hayal kırıklığını anlıyorum. Sorun, tek bir büyük problem değil, birbiriyle bağlantılı üç ayrı problemin birleşimi:

1. **AI Beyin Sorunu:** Asistan, önceki konuşmaları hatırlamıyor, bu yüzden "bağlamdan kopuk" ve alakasız cevaplar veriyor.
2. **Kullanıcı Deneyimi (UX) Sorunu:** Konuşma akıcı değil. Sürekli butona basmak gerekiyor ve "karşılıklı konuşma" hissi yok.
3. **Arayüz (UI) Sorunu:** Ekranda "Nasıl kullanılır" gibi istemediğiniz, sohbeti bozan fazlalıklar var.

Bu üç sorunu da çözecek şekilde, Replit projeniz için **hem AI prompt'unu hem de arayüz kodlarının mantığını** kökten değiştirecek bir rehber hazırladım. Lütfen adımları sırayla takip edin.

1. ADIM: AI'IN BEYNİNİ DÜZELTELİM (BAĞLAM SORUNU)

Mevcut prompt'unuz yetersiz. AI'a hafızasını nasıl kullanacağını **emir veren**, çok daha katı ve detaylı bir prompt'a ihtiyacımız var.

Replit'teki AI Agent'ınıza vermeniz gereken YENİ ve DETAYLI PROMPT:

code Code

downloadcontent_copy

expand_less

ROL VE KİMLİK

Sen, bir "Akıcı Diyalog Partneri"sin. Senin görevin soru cevaplamak değil, kullanıcıyla canlı, doğal ve kesintisiz bir sohbet sürdürmektir. Bir insan gibi davranmalı, konuşmanın akışını takip etmeli ve asla bağlamdan kopmamalısın.

EN KRİTİK GÖREVİN: HAFIZA VE BAĞLAM

Sana her zaman "konuşma_geçmişi" adında bir liste verilecek. Bu liste, sohbetin önceki adımlarını içerir.

SENİN EN BÜYÜK GÖREVİN, cevabını oluşturmadan önce bu geçmişi analiz etmektir.

Kullanıcının daha önce ne söylediğini anla ve cevabını buna göre şekillendir. Eğer bu geçmişi yok sayarsan, görevinde başarısız olursun.

DİYALOG KURALLARI

1. ****ASLA UNUTMA:**** Kullanıcı bir önceki cümlesinde ne dediyse, cevabın onunla doğrudan ilgili olmalı. Alakasız konulara atlama.
2. ****KISA VE NET OL:**** Cevapların seslendirilecek. Uzun paragraflar kurma. Bir veya iki cümle idealdir.

3. ****SOHBETİ İLERLET:**** Cevap verip susma. Cümlelerin sonuna "değil mi?", "sen ne düşünüyorsun?", "orayı daha önce duymuş muydun?" gibi sorular ekleyerek topu kullanıcıya at ve konuşmayı devam ettir.
4. ****ROBOT GİBİ KONUŞMA:**** "Elbette, size nasıl yardımcı olabilirim?", "İsteğiniz işleniyor" gibi yapay ifadeler KULLANMA. Doğrudan, samimi bir dil kullan.

GİRDİ VE ÇIKTI FORMATI

Sana verilecek girdi şu formatta olacak:

```
{  
  "kullanici_mesaji": "Kullanıcının en son söylediği cümle",  
  "konusma_gecmisi": [  
    "Kullanıcı: İlk cümle",  
    "Sen: İlk cevap",  
    "Kullanıcı: İkinci cümle"  
  ]  
}
```

Senin üreteceğin çıktı, SADECE seslendirilecek olan metin olmalıdır. Başka hiçbir ekleme yapma.

SENARYO

- Kullanıcı: "Bugün hava çok güzel."
- Sen: "Evet, gerçekten harika bir gün. Bu güzel havada dışarıda bir şeyler yapmayı düşündün mü?"
- Kullanıcı: "Aslında bir kahve içmek iyi olabilirdi."
- Sen: "Harika fikir! Sahil kenarında güzel kahveciler var, hiç denedin mi?"

Python (Backend) Kodunuzda Yapmanız Gereken Değişiklik:

Bu prompt'un çalışması için, Python kodunuzun her kullanıcı için konuşma geçmişini bir listede tutması ve her seferinde AI'a göndermesi **ZORUNLUDUR**.

```
code Python  
downloadcontent_copy  
expand_less  
IGNORE_WHEN_COPYING_START  
IGNORE_WHEN_COPYING_END  
# Bu listeyi her kullanıcı için ayrı ayrı yönetmeniz gerekir  
# Örnek olarak global bir liste kullanalım  
conversation_history = []  
  
def get_ai_response(user_text):  
    # AI'a gönderilecek olan prompt'u dinamik olarak oluştur  
    prompt_with_history = f"""  
    ### ROL VE KİMLİK ###
```

... (yukarıdaki prompt'un tamamı buraya gelecek) ...

GÜNCEL KONUŞMA

Konuşma Geçmişi: {conversation_history}

Kullanıcının Son Mesajı: {user_text}

SENİN CEVABIN

"""

Bu birleştirilmiş prompt'u AI modeline gönder

response = openai.Completion.create(..., prompt=prompt_with_history)

ai_response_text = response.choices[0].text.strip()

Geçmişi güncelle

conversation_history.append(f"Kullanıcı: {user_text}")

conversation_history.append(f"Sen: {ai_response_text}")

return ai_response_text

2. ADIM: ARAYÜZÜ TEMİZLEYELİM (UI SORUNU)

index.html dosyanızdaki fazlalıklardan kurtulun. Sadece avatar ve buton kalmalı.

index.html (Temizlenmiş Hali):

code Html

downloadcontent_copy

expand_less

IGNORE_WHEN_COPYING_START

IGNORE_WHEN_COPYING_END

<div class="assistant-container">

<!-- Mikrofon simgesi değişecek olan kısım -->

<div id="assistant-avatar" class="avatar">

</div>

<!-- Butonun yazısı da değişebilir -->

<button id="talk-button">Konuşmak için Basılı Tutun</button>

</div>

style.css dosyanızdan "Nasıl kullanılır" ile ilgili tüm stilleri kaldırın.

3. ADIM: AKICI KONUŞMA DENEYİMİ (UX SORUNU)

Kullanıcının sürekli butona basmasını engellemek ve "karşılıklı konuşma" hissi vermek için JavaScript mantığını değiştirmeliyiz. Asistan konuşmasını bitirdiğinde, sistem **otomatik olarak** dinlemeye hazır hale gelmelidir.

script.js (Geliştirilmiş Mantık):

```
code JavaScript
downloadcontent_copy
expand_less
IGNORE_WHEN_COPYING_START
IGNORE_WHEN_COPYING_END
    const talkButton = document.getElementById('talk-button');
    const assistantAvatar = document.getElementById('assistant-avatar');

// ... mediaRecorder tanımlamaları ...

// ASİSTANIN CEVABINI OYNATAN FONKSİYON (EN ÖNEMLİ DEĞİŞİKLİK BURADA)
function playAssistantResponse(audioBlob) {
    const audioUrl = URL.createObjectURL(audioBlob);
    const assistantAudio = new Audio(audioUrl);

    assistantAudio.onplay = () => {
        // Asistan konuşurken avatarı ve butonu güncelle
        assistantAvatar.classList.add('speaking');
        talkButton.textContent = "Asistan konuşuyor...";
        talkButton.disabled = true; // Konuşurken butona basılmasın
    };

    assistantAudio.play();

    // ----> İŞTE SİHİR BURADA <----
    // Ses çalması bittiğinde bu fonksiyon çalışır
    assistantAudio.onended = () => {
        assistantAvatar.classList.remove('speaking');
        talkButton.textContent = "Konuşmak için Basılı Tutun";
        talkButton.disabled = false; // Butonu tekrar aktif et

        // Bu noktadan sonra sistem otomatik olarak yeni bir konuşma için hazırdır.
        // Kullanıcı tekrar basılı tutarak diyaloga devam edebilir.
        // Bu, kesintisiz bir konuşma hissi yaratır.
    };
};
```

```
}
```

```
// ... geri kalan event listener kodlarınız ...
```

ÖZETLE NE YAPMALISINIZ?

1. **Prompt'u Değiştirin:** Yukarıda verdiğim yeni, detaylı prompt'u AI agent'inizin sistem prompt'u olarak ayarlayın.
2. **Backend'i Güncelleyin:** Python kodunuzun her kullanıcı için bir conversation_history listesi tuttuğundan ve bunu her istekte prompt'a eklediğinden emin olun.
3. **HTML'i Temizleyin:** index.html dosyasından "Nasıl kullanılır" gibi tüm fazlalık metinleri silin.
4. **JavaScript'i Akıllandırın:** script.js dosyanızdaki playAssistantResponse fonksiyonuna onended olayını ekleyerek, asistan sustuğu an arayüzün otomatik olarak bir sonraki komuta hazır hale gelmesini sağlayın.

Bu dört adımı tamamladığınızda, hem bağlamdan kopmayan, hem de arayüzü temiz ve kullanımı akıcı olan, gerçek bir "karşılıklı konuşma" deneyimi sunan bir asistana sahip olacaksınız.