

Anladım, şimdi projenin bir sonraki aşamasına geçiyoruz: Sadece bir sesli asistan değil, aynı zamanda görsel bir varlığı ve daha sezgisel kontrollere sahip, uygulama benzeri bir deneyim yaratmak.

Bu istekler, AI prompt'unun kendisinden çok, Replit'teki **kullanıcı arayüzü (Front-End)** ile ilgilidir. AI'nın ne söyleyeceğini belirleyen önceki prompt'umuz hala mükemmel, ancak şimdi bu çıktıyı kullanıcıya nasıl sunacağımızı ve kullanıcıdan nasıl girdi alacağımızı değiştireceğiz.

Bunun için Replit projenizde **HTML, CSS ve JavaScript** kullanmanız gerekecek. İşte bu sistemi kurmak için izlemeniz gereken adımlar ve mantık:

---

## Görsel ve İnteraktif AI Asistanı için Yapı

Bu yapıyı 3 ana bölümde ele alacağız:

1. **Görsel Avatar (HTML & CSS):** Asistanın "yüzü".
2. **Ses Kontrolleri (JavaScript):** Konuşmayı başlatma ve durdurma.
3. **Arka Plan Akışı (Backend & AI):** Mevcut sistemin bu yeni arayüze bağlanması.

---

### 1. Görsel Avatar (HTML ve CSS)

Amacımız, asistan konuşurken görsel bir geri bildirim vermek. ElevenLabs sitesindeki gibi dudak senkronizasyonu (lip-sync) çok ileri bir tekniktir, ancak çok benzer bir etkiyi basit animasyonlarla yaratabiliriz.

**index.html dosyanıza eklemeniz gereken temel yapı:**

```
code Html
downloadcontent_copy
expand_less
<!DOCTYPE html>
<html>
<head>
  <title>AI Asistan Yaratıcısı</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="assistant-container">
    <!-- Bu div, asistanın görseli olacak -->
    <div id="assistant-avatar" class="avatar"></div>
```

```
<!-- Konuşma ve Durdurma için tek bir buton -->
<button id="talk-button">Konuşmak için Basılı Tutun</button>
</div>

<!-- JavaScript dosyamızı en sona ekliyoruz -->
<script src="script.js"></script>
</body>
</html>
```

### **style.css dosyanızda avatarı ve animasyonu oluşturun:**

```
code Css
downloadcontent_copy
expand_less
IGNORE_WHEN_COPYING_START
IGNORE_WHEN_COPYING_END
body {
  background-color: #1a1a1a;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  font-family: sans-serif;
}

.assistant-container {
  text-align: center;
}

.avatar {
  width: 150px;
  height: 150px;
  background-color: #007bff;
  border-radius: 50%;
  margin: 0 auto 30px auto;
  transition: all 0.2s ease-in-out;
}

/* Asistan konuşurken bu class eklenecek */
.avatar.speaking {
  box-shadow: 0 0 40px 10px #007bff; /* Parlama efekti */
  transform: scale(1.05); /* Hafif büyüme efekti */
}
```

```
#talk-button {  
  padding: 15px 30px;  
  font-size: 18px;  
  cursor: pointer;  
  border: none;  
  border-radius: 50px;  
  background-color: #28a745;  
  color: white;  
}
```

---

## 2. Ses Kontrolleri ve Arayüz Mantığı (JavaScript)

Bu en önemli kısım. Kullanıcının butona basmasını dinleyecek, sesi kaydedecek ve asistan konuşurken avatarı canlandıracak kod.

**script.js dosyanızın içeriği bu mantığı izlemeli:**

```
code JavaScript  
downloadcontent_copy  
expand_less  
IGNORE_WHEN_COPYING_START  
IGNORE_WHEN_COPYING_END  
  // HTML elementlerini seçelim  
const talkButton = document.getElementById('talk-button');  
const assistantAvatar = document.getElementById('assistant-avatar');  
  
let mediaRecorder; // Ses kayıt nesnesi  
let audioChunks = []; // Kaydedilen ses parçaları  
let isAssistantSpeaking = false;  
let assistantAudio = null; // Asistanın sesini çalacak olan nesne  
  
// BUTONA BASILDIĞINDA (Konuşmayı Başlat)  
talkButton.addEventListener('mousedown', async () => {  
  if (isAssistantSpeaking) {  
    // Eğer asistan konuşuyorsa, butona basmak onu susturur  
    if(assistantAudio) {  
      assistantAudio.pause();  
      assistantAudio.currentTime = 0;  
    }  
    assistantAvatar.classList.remove('speaking');  
    isAssistantSpeaking = false;  
  } else {  
    // Asistan konuşmuyorsa, kullanıcı konuşmaya başlar
```

```

// Mikrofondan ses kaydını başlat
const stream = await navigator.mediaDevices.getUserMedia({ audio: true });
mediaRecorder = new MediaRecorder(stream);
mediaRecorder.start();

mediaRecorder.ondataavailable = event => {
  audioChunks.push(event.data);
};

talkButton.textContent = "Dinliyorum...";
}
});

// BUTON BIRAKILDIĞINDA (Konuşmayı Gönder)
talkButton.addEventListener('mouseup', () => {
  if (mediaRecorder && mediaRecorder.state === "recording") {
    mediaRecorder.stop();

    mediaRecorder.onstop = async () => {
      // Kaydedilen sesi bir dosyaya dönüştür
      const audioBlob = new Blob(audioChunks, { type: 'audio/webm' });
      audioChunks = []; // Listeyi temizle

      // Bu ses dosyasını backend'e (Python) gönder
      const response = await sendAudioToBackend(audioBlob);

      // Backend'den gelen cevabı (ses dosyası) oynat
      playAssistantResponse(response);
      talkButton.textContent = "Konuşmak için Basılı Tutun";
    };
  }
});

// Sesi Backend'e gönderme fonksiyonu
async function sendAudioToBackend(audioBlob) {
  const formData = new FormData();
  formData.append('audio', audioBlob);

  // Replit'teki Python sunucuna isteği gönder
  const response = await fetch('/process-audio', {
    method: 'POST',
    body: formData
  });
  // Cevap olarak gelen ses dosyasını al

```

```

    return await response.blob();
}

// Asistanın sesini oynatma fonksiyonu
function playAssistantResponse(audioBlob) {
    const audioUrl = URL.createObjectURL(audioBlob);
    assistantAudio = new Audio(audioUrl);

    // Ses çalmaya başladığında animasyonu başlat
    assistantAudio.onplay = () => {
        isAssistantSpeaking = true;
        assistantAvatar.classList.add('speaking');
        talkButton.textContent = "Durdurmak için Tıkla";
    };

    // Ses bittiğinde animasyonu durdur
    assistantAudio.onended = () => {
        isAssistantSpeaking = false;
        assistantAvatar.classList.remove('speaking');
        talkButton.textContent = "Konuşmak için Basılı Tutun";
    };

    assistantAudio.play();
}'''

```

---

### #### \*\*3. Arka Plan Akışı (Python Backend Değişikliği)\*\*

Python kodunuz artık metin değil, ses dosyası alacak. Bunun için bir "Speech-to-Text" (Sesi Metne Çevirme) servisi kullanmanız gerekir. (Örn: OpenAI Whisper, Google Speech-to-Text).

**\*\*Python (`main.py`) kodunuzdaki ilgili bölüm şöyle güncellenmeli:\*\***

```

'''python
from flask import Flask, render_template, request
import openai # veya hangi AI/STT servisini kullanıyorsanız

app = Flask(__name__)

# 1. Metni AI'a gönderip cevap alma (Bu fonksiyon zaten sizde var)
def get_ai_response(text):
    # ÖNCEKİ PROMPT'UNUZU BURADA KULLANIN
    # ...

```

```

# response_text = ...
return response_text

# 2. Metni ElevenLabs ile sese çevirme (Bu fonksiyon zaten sizde var)
def text_to_elevenlabs_audio(text):
    # ...
    # audio_data = ...
    return audio_data

# YENİ ADIM: Gelen sesi metne çevirme
def speech_to_text(audio_file):
    # Örneğin OpenAI Whisper kullanarak:
    transcript = openai.Audio.transcribe("whisper-1", audio_file)
    return transcript['text']

# ANA SAYFAYI GÖSTER
@app.route('/')
def index():
    return render_template('index.html')

# JAVASCRIPT'TEN GELEN SESİ İŞLEME
@app.route('/process-audio', methods=['POST'])
def process_audio():
    audio_file = request.files['audio']

    # 1. Gelen sesi metne çevir
    user_text = speech_to_text(audio_file)

    # 2. Metni AI Agent'a (prompt ile) gönder ve metin cevabı al
    ai_text_response = get_ai_response(user_text)

    # 3. AI'nın metin cevabını ElevenLabs ile sese çevir
    ai_audio_response = text_to_elevenlabs_audio(ai_text_response)

    # 4. Bu sesi Front-End'e geri gönder
    return ai_audio_response, 200, {'Content-Type': 'audio/mpeg'}

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=81)

```

## Özetle Ne Yaptık?

1. **Görünüm:** Kullanıcı artık yazı yazmıyor, sadece parlayan bir daire (avatar) ve bir buton görüyor.
2. **Girdi:** Kullanıcı butona basılı tutarak konuşuyor. JavaScript bu sesi kaydediyor.
3. **İletişim:** Kaydedilen ses, Python backend'ine gönderiliyor.
4. **İşleme:** Backend, sesi metne çeviriyor, bu metni sizin harika AI prompt'unuzla işliyor, gelen cevabı ElevenLabs ile sese dönüştürüyor.
5. **Çıktı:** Oluşturulan ses dosyası tarayıcıya geri gönderiliyor. JavaScript bu sesi çalarken aynı anda avatarı canlandırıyor.
6. **Kontrol:** Kullanıcı, asistan konuşurken butona tekrar basarak onu susturabiliyor.