



SAKARYA
ÜNİVERSİTESİ

Sakarya Üniversitesi

Bilgisayar ve Bilişim Bilimleri

Bilgisayar Mühendisliği Bölümü Yapay Zeka

2024-2025 Bahar Dönemi

Proje Adı: Yapay Zekâ Tabanlı Sesli Asistan

Hazırlayanlar

Ali Kerem Kol/ B221210042

Muhammed Baha Bakan/ B221210050

Yiğit Özpolat/ B221210004

Ders Grubu: 1/B

Ödev Tanıtım Linki:

<https://drive.google.com/file/d/140KwmFL63sbloch6C-QK4QeeDU2m2Bhv/view>

Bu projenin amacı, bir bireyin ve yaşadığı şehrin tanıtımını yapabilen, yapay zekâ destekli sesli asistan ile zenginleştirilmiş bir web tabanlı sistem geliştirmektir. Proje, ziyaretçilere hem bireysel hem de şehir hakkında bilgi sunmayı hedeflemekte, kullanıcı deneyimini etkileşimli ve kişiselleştirilmiş bir hale getirmektedir.

Tanıtımı yapılan şehir **Denizli** olup, bu sistem aracılığıyla Denizli'nin tarihi, kültürel, coğrafi ve turistik özellikleri kullanıcıya sesli olarak aktarılabilir. Bu tür bir sistem, şehir tanıtımı ve bireysel portfolyo sunumu gibi alanlarda interaktif çözümler sunmayı amaçlamaktadır.

Geliştirilen program, HTML, CSS ve JavaScript teknolojileri kullanılarak hazırlanmış bir web arayüzüne sahiptir. Bu arayüz üzerinden kullanıcı, sistemle etkileşime geçebilir. Programın temel özellikleri şunlardır:

- Kullanıcı, sayfaya girdiğinde bir sesli asistan ile karşılaşır.
- Sesli asistan, önceden tanımlanmış komutlara ve içeriklere bağlı olarak hem kullanıcının kendisi hem de Denizli şehri hakkında bilgi verir.
- Web arayüzü üzerinden sesli asistanla etkileşime geçmek kolaydır; kullanıcı komut vermese bile sistem belirli bilgileri otomatik olarak sunar.

Sesli Komut ile Cevap Üretme Süreci

- **Webhook**
Tarayıcıdan gelen ses verisini alır.
"Kullanıcının sesli komutu bu noktada sistemimize geliyor."
- **OpenAI (Transcribe Recording)**
Ses kaydını metne çevirir (Whisper).
"Ses verisi OpenAI'nın transkripsiyon servisiyle yazıya dökülüyor."
- **AI Agent (Tools Agent)**
Yazılı girdiyi analiz eder, gerekirse araçları kullanır.
"Bu düğüm gelen metni analiz eder ve uygun aracı seçerek yanıt üretir."
- **OpenAI Chat Model**
Chat tabanlı yanıt üretir (gelen metne göre).
"Temel sorular bu model üzerinden yanıtlanır."
- **Simple Memory**
Önceki mesajları hafızada tutar.
"Konuşmalar arası bağ kurabilmek için hafıza işlevi görür."
- **Calculator**
Matematiksel hesaplama yapar.
"Sayısal sorular bu düğümde çözülür."
- **Answer with Vector Store**
Dokümanlardan bilgi getirir (RAG).
"Daha detaylı bilgiler varsa vektör veritabanından yanıt alınır."

- **Pinecone Vector Store1**
Önceden yüklenmiş belgeleri saklar ve sorgular.
"Bu, yapay zekanın bilgi aradığı özel hafızadır."
- **OpenAI Chat Model1**
Vektör sonucu için ek doğal dil yanıtı üretir.
"Bulunan bilgileri insan diliyle özetler."
- **Embeddings OpenAI**
Metinleri vektöre çevirir (embedding).
"Verileri anlamlı hale getirmek için vektörleştirir."
- **OpenAI (Generate Audio)**
Üretilen metni sesli yanıtla çevirir (TTS).
"Gelen yanıt ses dosyasına çevrilir."
- **Respond to Webhook**
MP3 dosyasını frontend'e geri yollar.
"Son olarak yanıt tarayıcıya sesli olarak geri gönderilir."
- ---

Doküman Yükleme ve Embedding İşlemi

- **Google Drive Trigger**
Yeni dosya geldiğinde süreci başlatır.
"Drive'a yeni bir dosya yüklendiğinde bu süreç tetiklenir."
- **Google Drive (Download file)**
Dosyayı indirir.
"Yüklenen dosya buradan sistemimize çekilir."
- **Default Data Loader**
Dosyayı okur, parçalar ve işler.
"Bu adım dokümanı satır satır analiz eder."
- **Recursive Text Splitter**
Metni küçük parçalara ayırır.
"Veriyi daha iyi anlayabilmek için bölümlere ayırıyoruz."
- **Embeddings Cohere**
Parçaları vektöre çevirir.
"Her parçayı matematiksel temsillere dönüştürüyoruz."
- **Pinecone Vector Store**
Embed edilmiş verileri kaydeder.
"Tüm bu veriler özel veritabanımıza kaydedilir."

Kod Bölümü

Aşağıda yapay zeka sesi asistanının bulunduğu sayfanın kaynak kodları belirtilmiştir.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>ŞEHRİM</title>

<link rel="stylesheet" type="text/css" href="/CSS/stil.css">

<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,300;1,
300;1,400&display=swap" rel="stylesheet">

<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,300;1,
300;1,400&family=Teko:wght@500&display=swap" rel="stylesheet">

<link
href="https://fonts.googleapis.com/css2?family=Kanit:wght@300&display=swap"
rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=Alata&display=swap"
rel="stylesheet">

</head>

<body>

<nav>

<ul>

<li style="float:left"><a href="index.html" class="active"><i class="bi bi-
house"></i>HAKKIMDA</a></li>

<li><a href="LOGIN.html"><i class="bi bi-door-open-fill"></i>LOGİN</a></li>

<li><a href="İLETİŞİM.html"><i class="bi bi-blockquote-
left"></i>İLETİŞİM</a></li>

<li><a href="MİRASIMIZ.html"><i class="bi bi-tree"></i>MİRASIMIZ</a></li>

<li><a href="ŞEHRİM.html"><i class="bi bi-signpost-2"></i>ŞEHRİM</a></li>

<li><a href="HOBİLERİM.html"><i class="bi bi-code-
square"></i>HOBİLERİM</a></li>

<li><a href="ÖZGEÇMİŞ.html"><i class="bi bi-file-
person"></i>ÖZGEÇMİŞ</a></li>

</ul>

</nav>
```

```
<div id="container">
```

```
<header>
```

```

```

```
<div id="centered"> ŞEHRİM </div>
```

```
</header>
```

```
<section>
```

```
<main>
```

```
<h1>ŞEHRİM</h1>
```

```
<div class="anakisim">
```

```

```

```
<h3 id="denizliYazi">DENİZLİ</h3>
```

```
<div><h4 id="nüfus">NÜFUS: 1.051.056</h4></div>
```

```
<div class="sehrimyazi"><h1>-TRAVERTENLER-</h1><p>Menderes Nehri  
Vadisinde bulunur...</p></div>
```

```
<div class="sehrimyazi"><h1>-LAODİCEA-</h1><p>Kentin adı antik  
kaynaklarda daha çok...</p></div>
```

```
<div class="sehrimyazi"><h1>-İNCEĞİZ KANYONU-</h1><p>Tamamen  
doğal oluşuma sahip kanyon...</p></div>
```

```
<div class="sehrimyazi"><h1>-HİERAPOLİS-</h1><p>Hierapolis Antik  
Kenti'nin en büyük yapılarından biri...</p></div>
```

```
</div>
```

```
<div class="cerceve">
```

```
<div class="galeri">
```

```
<a href="https://tr.wikipedia.org/wiki/Pamukkale" target="_blank"></a>
```


</div>

</div>

<!-- 🎤 Voice Assistant Section -->

<div id="voiceAssistant" style="text-align:center; margin: 60px 0;">

<h2> 🎤 Yapay Zekâ Asistanı</h2>

<button id="recordButton" type="button" style="padding: 15px 30px; font-size: 16px;">Konuşmaya Başla</button>

<div id="transcript" style="margin-top: 20px; font-size: 18px;"></div>

<div id="response" style="margin-top: 20px; font-size: 18px;"></div>

<audio id="audioPlayer" controls style="display:none; margin-top: 20px;"></audio>

<div id="downloadLink" style="margin-top: 20px; font-size: 18px;"></div>

</div>

</main>

</section>

<footer>

Tüm hakları saklıdır © |2023

</footer>

</div>

```
<!-- Voice Assistant Script -->
```

```
<script>
```

```
console.log("🔗 Sayfa yüklendi.");
```

```
const recordButton = document.getElementById("recordButton");
```

```
const transcriptDiv = document.getElementById("transcript");
```

```
const responseDiv = document.getElementById("response");
```

```
const audioPlayer = document.getElementById("audioPlayer");
```

```
const downloadLinkDiv = document.getElementById("downloadLink");
```

```
let mediaRecorder;
```

```
let audioChunks = [];
```

```
async function sendToWebhook(audioBlob) {
```

```
  try {
```

```
    console.log("📡 Webhook'a veri gönderiliyor...");
```

```
    const formData = new FormData();
```

```
    formData.append("audio", audioBlob, "recording.wav");
```

```
    const webhookUrl = "http://localhost:5678/webhook/voice-input";
```

```
    const response = await fetch(webhookUrl, {
```

```
      method: "POST",
```

```
      body: formData
```

```
    });
```

```
    if (!response.ok) throw new Error(🌀 HTTP error! status: ${response.status});
```

```
    const audioBlobResponse = await response.blob();
```

```
const audioUrl = URL.createObjectURL(audioBlobResponse);
```

```
audioPlayer.src = audioUrl;
```

```
audioPlayer.style.display = "block";
```

```
try {
```

```
  await audioPlayer.play();
```

```
  console.log("✅ Ses başarıyla çalındı.");
```

```
} catch (err) {
```

```
  console.error("❌ Ses çalarken hata:", err);
```

```
}
```

```
downloadLinkDiv.innerHTML = ▼ <a href="${audioUrl}"  
download="response.mp3">Yanıtı indir</a>;
```

```
responseDiv.textContent = "🔊 Yanıt seslendiriliyor...";
```

```
} catch (err) {
```

```
  console.error("💩 Bir hata oluştu:", err);
```

```
  responseDiv.textContent = "💩 Bir hata oluştu. Konsolu kontrol et.";
```

```
}
```

```
}
```

```
recordButton.addEventListener("click", async (event) => {
```

```
  event.preventDefault();
```

```
  if (!navigator.mediaDevices) {
```

```
    alert("Microphone access is not supported");
```

```
    return;
```

```
  }
```



```
const stream = await navigator.mediaDevices.getUserMedia({ audio: true });
mediaRecorder = new MediaRecorder(stream);
audioChunks = [];

mediaRecorder.ondataavailable = event => {
  if (event.data.size > 0) audioChunks.push(event.data);
};

mediaRecorder.onstop = async () => {
  const audioBlob = new Blob(audioChunks, { type: "audio/wav" });
  transcriptDiv.textContent = "📎 Gönderiliyor...";
  await sendToWebhook(audioBlob);
};

mediaRecorder.start();
recordButton.textContent = "Kaydediliyor...";

setTimeout(() => {
  mediaRecorder.stop();
  recordButton.textContent = "Konuşmaya Başla";
}, 5000);
});
</script>

</body>
</html>
```