



T.C
KIRKLARELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ

MUF13402 – BİTİRME PROJESİ II
Dr. Öğr. Üyesi Edip Serdar GÜNER

OTOMATİK KONUŞMA TANIMA

1170505001 – Tansu ÇAM
1160505002 – Burak BİLGİN
1170505015 – Hakan KÖKCEN

İçindekiler

1. Giriş	3
2. Kullanıcı İşlemleri	4
2.1. Kaydolma	5
2.2. Giriş.....	6
2.3. Token Satın Alma	7
2.4. Dosya Yükleme ve Görüntüleme.....	8
2.5. Transkripsiyon	10
2.6. Dosya Silme.....	11
3. Yönetici İşlemleri	12
3.1. Dosyaları Görüntüleme	12
4. Örnek Kodlar	14
4.1. Örnek Component	14
4.2. Örnek Page Dosyası	14
4.3. Örnek API Dosyası	15
4.4. Örnek Model Dosyası	16
4.5. Google Speech-To-Text API'ı İle Bağlantı.....	16

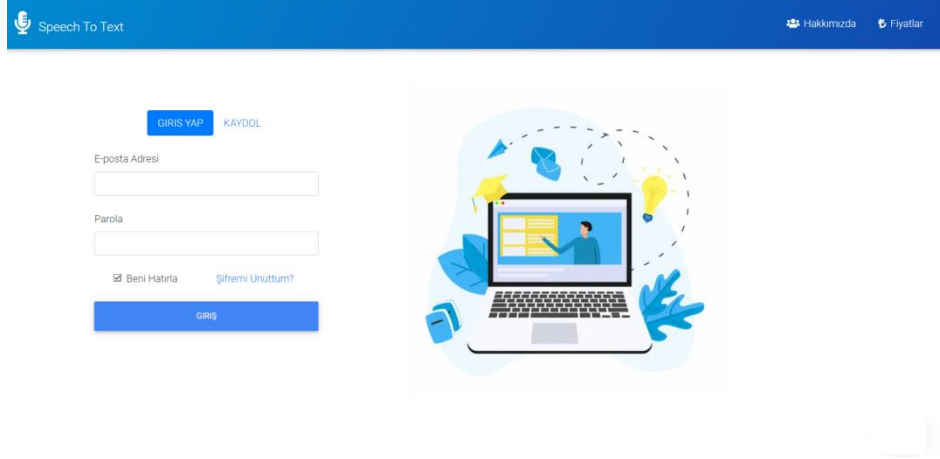
1. Giriş

Projede Google yapay zeka teknolojilerinin sunduğu ses tanıma API'yı kullanılmıştır. Projenin ön yüzünde, arayüz oluşturmak için React ve NextJS kütüphaneleri, arka yüzünde ağ tabanlı uygulamalar geliştirmek için tasarlanmış olan Node.JS çalışma ortamı ve verilerin tutulması, hızlı olarak işlenmesi için NoSQL bir veritabanı uygulaması olan MongoDB (Mongoose nesne modeli) kullanılmıştır.

Projede transkripsiyon işleminden faydalanan kullanıcı ile fiyatlandırma ve işlem yönetimi işini gerçekleştiren yönetici olmak üzere 2 temel aktör bulunmaktadır. Ziyaretçi olarak transkripsiyon işlemi gerçekleştirilememektedir. Ziyaretçi web sitesini açtığında veya kayıttan önce herhangi bir işlem yapmaya çalıştığında kayıt sayfasına yönlendirilir.

Transkripsiyon işlemi asenkron olarak Google Speech-To-Text API aracılığıyla gerçekleştirilmektedir. Kullanıcılar Türkçe dilinde, noktalama işaretlerine uygun olarak yüksek doğruluk oranıyla transkript alabilmektedir. Transkripsiyon işlemini gerçekleştirmeden önce ilk adımda dosya sisteme yüklenir, ses dosyasının uzunluğuna göre ücret belirlenir. Kullanıcının bakiyesinde yeterli token bulunuyorsa transkript alma işlemi için komut verebilir aksi halde işlem gerçekleştirilmez. Kullanıcı işlenmesini istemediği veya işlemi bitmiş dosyaları sistemden silebilir.

2. Kullanıcı İşlemleri



Şekil 1: Ana sayfa

Ziyaretçi giriş yapmadığı sürece Şekil 1'deki ekran ile karşılaşmaktadır. Ziyaretçi eğer hesabı varsa giriş yapabilir, yoksa kaydolma bölümünden yeni kayıt oluşturabilir. Veya giriş yapma niyeti yoksa hakkımızda ve fiyatlar bölümünden hizmet ile ilgili bilgi alabilir. Eğer bir hesabı varsa ve şifresini bilmiyorsa, şifremi unuttum bölümünden şifresini yenileyebilir.

2.1. Kaydolma

[GİRİS YAP](#) [KAYDOL](#)

Ad

Soyad

E-posta Adresi

Parola

[KAYDOL](#)

Şekil 2: Kaydolma bölümü

Ziyaretçinin herhangi bir hesabı yoksa ilgili alanları doldurarak sisteme giriş yapabilir. Proje canlıda olmadığı için herhangi bir mail hizmeti kullanılmamıştır, kaydol butonuna tıkladıktan sonra hesap otomatik olarak sisteme eklenmektedir.

2.2. Giriş

GİRİŞ YAP

KAYDOL

E-posta Adresi

hakankoccen@s2t.com

Parola

.....

☒ Beni Hatırla

[Şifremi Unuttum?](#)

GİRİŞ

Şekil 3: Giriş yapma

Şekil 3’te de görüldüğü gibi, kaydolurken kullanılan e-posta adresi ve şifre ile eğer ilgili kullanıcı mevcutsa veya sistemden silinmediyse giriş yapılabilir.

Speech To Text

HakkımızdaFiyatlarProfil

Dosya Yükle

Dosya Yükle


Ses ya da video dosyanızı mp3/mp4 formatında yükleyiniz.

Yükle

Dosya Adı

Browse

GÖNDER

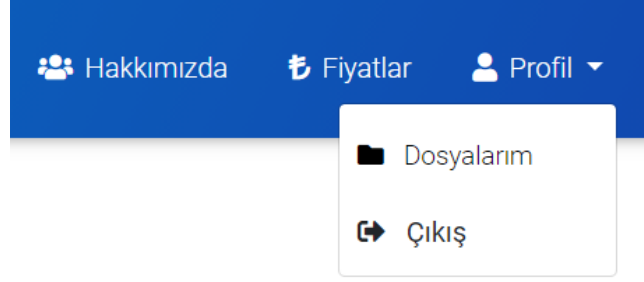


Daha Doğru Sonuçlar İçin

- Gürültüsüz ses dosyaları kullanın.
- Aynı anda konuşulmayan dosyalar kullanın.
- Aksansız konuşulan dosyaları tercih edin.

Şekil 4: Giriş işleminden sonra gelen ekran

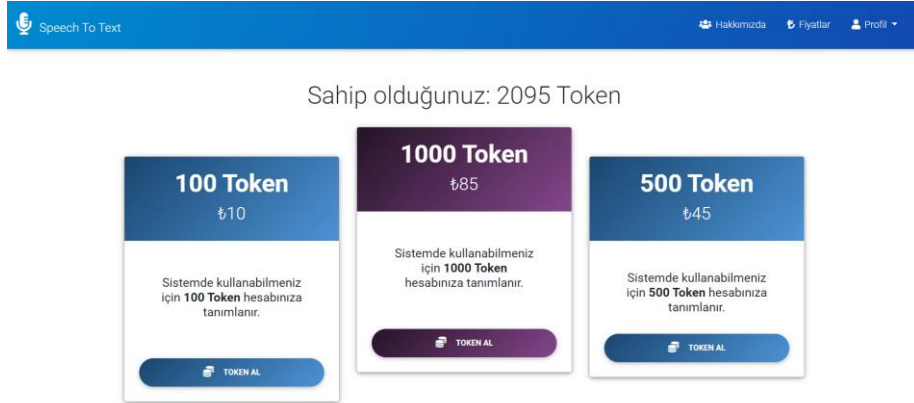
Giriş işlemi başarılı ise artık ana sayfa olarak Şekil 4'teki dosya yükleme ekranı gelecektir. Kullanıcı dilerse header'daki navigasyon bölümünden hakkımızda ve fiyatlar sekmesine giderek bilgi alabilir veya profil çekmecesinden daha önceki dosyalarını görüntüleyebilir veya çıkış yapabilir.



Şekil 5: Navigasyon

Kullanıcı eğer çıkış yaparsa tekrar, ziyaretçilerin gördüğü giriş sayfasına yönlendirilir. (Şekil 1)

2.3. Token Satın Alma

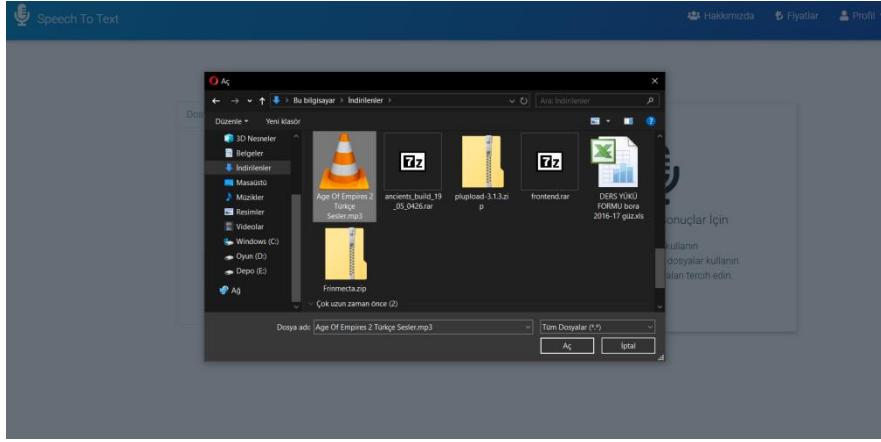


Şekil 6: Fiyatlar sayfası

Kullanıcının transkripsiyon işlemi gerçekleştirebilmesi için, dosya uzunluğuna göre belirlenmiş belli bir miktar tokene sahip olması gerekmektedir. İlk giriş yapan kullanıcının hesabında hiç token

bulunmaz fakat kullanıcı fiyatlar sekmesinden kendine uygun paketi seçerek token alabilir. Proje canlıda olmadığı için herhangi bir ödeme hizmeti kullanılmamıştır, “Token Al” butonuna tıklanarak istenilen paket hesaba tanımlanabilir.

2.4. Dosya Yükleme ve Görüntüleme



Şekil 7: Dosya yükleme penceresi

Kullanıcı desteklenen ses dosyalarını karşılayan bir dosyayı seçerek sisteme yükleyebilir.

Dosya Yükle

Dosya Yükle

Ses ya da video dosyanızı mp3/mp4 formatında yükleyiniz.

Yükle

Age Of Empires 2 Türkçe Sesler.mp3

Browse

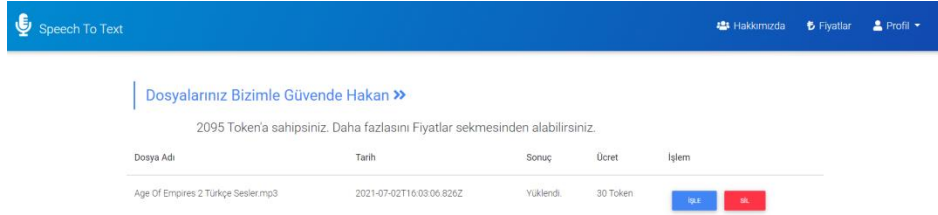
GÖNDER

Şekil 8: Yükleme işlemini onaylama

Dosya seçildikten sonra kullanıcı eğer yanlış dosya seçtiyse “Browse” butonuna tıklayarak tekrar dosya seçebilir veya dosya doğru ise “Gönder” butonuna tıklayarak seçimini onaylayabilir. (Şekil 8)

Bu aşamadan sonra dosya yükleme işlemi başlayacak ve kullanıcı dosya büyüklüyle orantılı olarak bir süre bekletilecektir. İşlem bittikten sonra kullanıcı dosyalarım sayfasına yönlendirilecektir.

Dosya yükleme işlemi esnasında herhangi bir ücret ödenmemektedir.



Dosya Adı	Tarih	Sonuç	Ücret	İşlem
Age Of Empires 2 Türkiye Sesler.mp3	2021-07-02T16:03:06.826Z	Yüklendi	30 Token	<button>İşle</button> <button>Sil</button>

Şekil 9: Dosyalarım sayfası

Kullanıcı dosyalarım sayfasında (Şekil 9), yükleme tarihine göre sıralanmış şekilde yüklemiş olduğu dosyalarını görüntüleyebilir. Son yüklenen dosya en başta sıralanmaktadır. Tablodaki dosya adı alanı dosyanın adını ve uzantısını, tarihi yükleme tarih ve saatini, sonuç alanı dosyanın şu anki durumunu ve ücret alanı da transkripsiyon için gereken tokeni göstermektedir. Her dosyanın işlem ücreti, dosyanın uzunluğuyla doğru orantılıdır.

2.5. Transkripsiyon

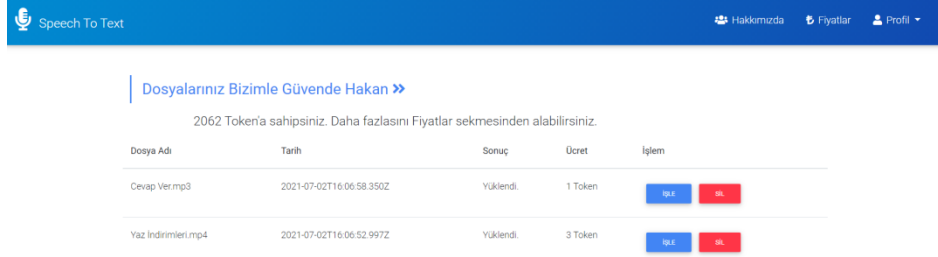
Speech To Text					
			Hakkımızda	Fiyatlar	Profil
Dosyalarınız Bizimle Güvende Hakan >>					
2062 Token'a sahipsiniz. Daha fazlasını Fiyatlar sekmesinden alabilirsiniz.					
Dosya Adı	Tarih	Sonuç	Ücret	İşlem	
Cevap Ver.mp3	2021-07-02T16:06:58.350Z	Yüklendi.	1 Token	İşle	İs
Yaz İndirimleri.mp4	2021-07-02T16:06:52.997Z	Yüklendi.	3 Token	İşle	İs
Age Of Empires 2 Türkçe Sesler.mp3	2021-07-02T16:03:06.826Z	Efendim efendim efendim emrinden rin Evet evet evet Hazır hazır. Madenci Oduncu Oduncu Allah Allah Avcı Avcı Balıkçı Tamamı hamamı tamirci Usta Yapanım yaparım.	30 Token	İşle	İs

Şekil 10: Transkripsiyonun sonucu

Kullanıcı eğer tokeni yetiyorsa işlem yapmak istediği dosyaya ait satırın “İşle” butonuna tıklayarak transkripsiyon isteği gönderebilir. Sistem kullanıcı bir süre beklettikten sonra sonuç alanı güncellenmiş bir şekilde sayfayı yenileyecektir. Bekleme süresi, dosya boyutuna göre değişiklik göstermektedir.

Şekil 10’da da görüldüğü gibi sonuç bölümüne transkripsiyon sonrası üretilmiş metnin kendisi yazılmaktadır.

2.6. Dosya Silme



The screenshot shows the 'Speech To Text' application interface. At the top, there is a blue header bar with the application name and navigation links. Below the header, a message indicates the user's token balance and provides a link to view more details. The main content area features a table with columns for file name, date, status, cost, and actions. Two files are listed: 'Cevap Ver.mp3' and 'Yaz İndirimleri.mp4'. Each file row has a 'İşle' (Action) button with a 'Sil' (Delete) option.

Dosya Adı	Tarih	Sonuç	Ücret	İşlem
Cevap Ver.mp3	2021-07-02T16:06:58.350Z	Yüklendi.	1 Token	İşle Sil
Yaz İndirimleri.mp4	2021-07-02T16:06:52.997Z	Yüklendi.	3 Token	İşle Sil

Şekil 11: Silme işlemi sonrası yeni tablo

Kullanıcı silmek istediği dosyanın olduğu satırdaki “Sil” butonuna tıklayarak ilgili dosyayı sistemden silebilir. Dosya silindikten sonra token iade edilmez ve geçmişi tutulmaz. Şekil 11’de görüldüğü üzere alan silinerek diğer kayıtlar üzerinde işlem yapılmaya devam edilebilir.

4. Örnek Kodlar

4.1. Örnek Component

Ön yüzdeki grafik bileşenlerinin React kütüphanesi sayesinde component sınıfı ya da fonksiyonu olarak yazılarak istenildiğinde tekrar tekrar kullanılması ve ayrı bir dosyada bulunmasını sağlar.

```
components > Footer.js > default
1  const Footer = () =>{
2    return(
3      <footer class="text-center text-white" style={{backgroundColor: '#0a4275',position:'static',left:0,bottom:0,width:'100%'}}>
4
5
6      <div class="text-center p-3" style={{backgroundColor: 'rgba(0, 0, 0, 0.2)}}>
7        © 2021 Copyright:
8        <span class="text-white" > SPTT.com</span>
9      </div>
10
11    </footer>
12  )
13 }
14 export default Footer
```

Şekil 14: Footer Component'i

4.2. Örnek Page Dosyası

NextJS kütüphanesi ile pages klasöründe oluşturulan her dosya için otomatik olarak routing işlemlerin hazırlanır. Dosya içeriğinde kullanılan componentler, sayfaya özel yazılan JSX kodları ile sayfa render edilir.

```
pages > price.js > ...
1  import React from 'react'
2  import {parseCookies} from 'nookies'
3  import PriceCards from '../components/Price'
4  import User from '../models/User'
5
6  const Price = (props) => {
7    return (
8      <PriceCards props={props} />
9    )
10  }
11
12  export async function getServerSideProps(ctx){
13    const cookie = parseCookies(ctx)
14    const user = cookie.user ? JSON.parse(cookie.user) : ""
15    if(user.role == 'admin'){
16      const {res} = ctx
17      res.writeHead(302,{Location:"/"})
18      res.end()
19    }
20    const userToken = await User.findById(user._id)
21    return {
22      props: {
23        id: user._id,
24        tokenAmount: userToken.token,
25      }
26    }
27  }
28
29  export default Price
```

Şekil 15: Fiyatlar ve Token Alma Sayfası

4.3. Örnek API Dosyası

Arka plan işlemleri için pages/api klasöründe bulunan NodeJS üzerinde çalışan JS kodlarını içeren dosyalardır. NextJS ile pages klasöründeki ön yüz sayfaları gibi otomatik olarak route ayarları yapılır ve API endpointleri oluşur.

```
pages > api >  transFile.js >  default
1  import User from "../../models/User";
2  import File from "../../models/File";
3  import initDB from "../../helpers/initDB";
4  import googleSTT from "../../helpers/googleSTT";
5
6  initDB()
7  export default async (req,res)=>{
8    const {fileId, userId} = req.body;
9
10   const user = await User.findById(userId);
11   const file = await File.findById(fileId);
12
13   if (file.duration > user.token) {
14     res.json({ error: "Yetersiz token." })
15   } else {
16     user.token -= file.duration;
17     await User.updateOne({_id: userId}, user);
18     const result = googleSTT(res, file);
19   }
20 }
```

Şekil 16: Kullanıcı yüklediği dosyayı token'i yeterli ise transkript işlemini başlatan endpoint dosyası

4.4. Örnek Model Dosyası

MongoDB üzerindeki database collection'larının bir şemasını oluşturan dosyalardır.

```
models > File.js > fileSchema
1 import mongoose from 'mongoose'
2
3 const fileSchema = new mongoose.Schema({
4   name: {
5     type: String,
6     required: true
7   },
8   path: {
9     type: String,
10    required: true
11  },
12  transcript: {
13    type: String,
14    default: "Yüklendi."
15  },
16  hidden: Boolean,
17  user: {
18    type: mongoose.Schema.Types.ObjectId,
19    ref: 'User'
20  },
21  duration: {
22    type: 'Number',
23    required: true
24  },
25 }, {
26   timestamps: true
27 })
28
29 module.exports = mongoose.models.File || mongoose.model('File', fileSchema);
```

Şekil 17: Yüklenen dosyaların kaydedildiği collection şeması

4.5. Google Speech-To-Text API'ı İle Bağlantı

Sistemimize yüklenen ses dosyalarının Google Cloud hizmetlerinden olan Bucket'a upload edilmesi ve ardından yine Google Cloud hizmeti olan Speech-To-Text istemcisine transkripsiyon emri verilmesi. Gelen cevabın veritabanına kaydedilmesi işlemlerinin yapıldığı dosya.


```

helpers > googleSTT.js > ...
1 require('dotenv').config({ path: '../.env.local' });
2 const _ = require('lodash');
3 const speech = require('@google-cloud/speech');
4 const {Storage} = require('@google-cloud/storage');
5 const fs = require('fs');
6 const path = require('path');
7 const File = require('../models/File');
8
9 const googleSTT = (res, file) => {
10
11     const speechClient = new speech.SpeechClient();
12
13     const bucketName = 's2text-bucket';
14
15     const uploadToGcs = async () => {
16         const storage = new Storage();
17
18         const bucket = storage.bucket(bucketName);
19         const fileName = path.basename(file.path);
20
21         await bucket.upload(file.path);
22         return `gs://${bucketName}/${fileName}`;
23     };
24
25     uploadToGcs()
26     .then(async (gcsUri) => {
27         const audio = {
28             uri: gcsUri,
29         };
30
31         const config = {
32             encoding: 'mp3',
33             sampleRateHertz: 24000,
34             languageCode: 'tr-TR',
35             enableAutomaticPunctuation: true,
36         };
37
38         const request = {
39             audio,
40             config,
41         };
42
43         speechClient.longRunningRecognize(request)
44         .then((data) => {
45             const operation = data[0];
46
47             return operation.promise();
48         })
49         .then((data) => {
50             const results = _.get(data[0], 'results', []);
51             const transcription = results
52                 .map(result => result.alternatives[0].transcript)
53                 .join('\n');
54             console.log(`Transcription: ${transcription}`);
55             File.updateOne({_id: file._id}, {transcript: transcription}, (err) => {
56                 if (err) res.status(500).json(err);
57                 else res.status(200).json({message: 'File transcribed'});
58             });
59         })
60     })
61     .catch(err => {
62         console.error('ERROR:', err);
63     });
64
65 };
66
67 module.exports = googleSTT;

```

Şekil 18: googleSTT.js dosya içeriği