



T.C
HALIÇ ÜNİVERSİTESİ
MESLEK YÜKSEKOKULU
BİLGİSAYAR TEKNOLOJİSİ BÖLÜMÜ

ENGELLİ BİREYLER İÇİN SESLİ UYGULAMA (VOICEBOT)

ÖN LİSANS TEZİ

Hazırlayan
Berfin Aleyna SUKUTİ

Danışman
Neşe Aslı KÖSE

İstanbul – 2023

T.C
HALIÇ ÜNİVERSİTESİ
MESLEK YÜKSEKOKULU
BİLGİSAYAR TEKNOLOJİSİ BÖLÜMÜ

ENGELLİ BİREYLER İÇİN SESLİ UYGULAMA
(VOICEBOT)

ÖN LİSANS TEZİ

Hazırlayan
Berfin Aleyna SUKUTİ

Danışman
Neşe Aslı KÖSE

İstanbul – 2023

ÖN SÖZ

Hazırladığım bu proje; Haliç Üniversitesi, Bilgisayar Teknolojisi bölümü 2021-2023 dönemleri bitirme projesidir. Dördüncü dönemde hazırlanmış ve jüriye sunulmuştur.

Yaptığım projede öncelikle danışman hocalarım olan Ebru İnan BARUTCU ve Neşe Aslı KÖSE hocalarıma teşekkür ediyorum. Emeği geçen diğer hocalarıma da teşekkürlerimi sunuyorum.

İstanbul, 2023

Berfin Aleyna SUKUTİ

İÇİNDEKİLER

	Sayfa No.
KISALTMALAR.....	VI
ŞEKİLLER	VII
GENEL BİLGİLER ve ÖZET	VIII
GENERAL INFORMATION & ABSTRACT	IX
1. GİRİŞ	1
1.1 VoiceBot Masaüstü Uygulama	1
1.2 VoiceBot Masaüstü Uygulama Avantajları	2
1.3 VoiceBot Masaüstü Uygulamasını Destekleyen Platformlar	2
1.4 Microsoft Windows İşletim Sistemi	3
1.5 Windows İşletim Sisteminin Avantajları	3
1.6 Google Play	4
1.7 Masaüstü Uygulama	4
1.7.1 Masaüstü Uygulama Avantajları	4
1.7.2 Masaüstü Uygulama Geliştirme	5
1.8 Literatür Taraması	5
1.8.1 Dragon Dictation: İşitme Engelliler İçin İletişim	6
1.8.2 JAWS Yazılım Uygulaması	7
1.8.3 CEYDA: Engelleri Aşan Uygulama	8
2. AMAÇ	10
3. GEREÇLER	11
3.1 Visual Studio Code Nedir	12
3.2 Python Nedir	13
3.2.1 Python Dilinin Avantajları	14
3.2.2. Python Kütüphaneleri	14
3.3 SQLite Nedir	15
3.3.1 SQLite Yapısı ve Komutları	16
4. YÖNTEMLER	17
4.1 Kayıt Ekranı (Başlangıç Ekranı) Arayüzü	19
4.2 Giriş Ekranı Arayüzü	24
4.3 Ana Ekran Arayüzü	28
4.3 Uygulamamın Veri Tabanı Kullanıcılar.db	38
4.4 Tablo Oluşturma Kodları	39
4.5 Tabloya Veri Ekleme Kodları	40

4.6 Parola Deęiřtirme Kodu	41
4.7 Uygulamadan Hesap Silme Kodu	42
4.8 Tablo Silme Kodu.....	43
5. SONUÇ.....	44
6. TARTIřMA.....	45

KISALTMALAR

VSCode : Visual Studio Code

IDE : Integrated Development Environment (Entegre Geliştirme Ortamı)

ŞEKİLLER

Sayfa No.

Şekil 1:Dragon Dictation.....	6
Şekil 2: JAWS Yazılım.....	7
Şekil 3: CEYDA Sesli Asistan	8
Şekil 4: Visual Studio Code	12
Şekil 5: Python.....	13
Şekil 6: SQLite Veri Tabanı.....	15
Şekil 7: VoiceBot Kayıt Ekranı	19
Şekil 8: VoiceBot Giriş Ekranı	24
Şekil 9: VoiceBot Başarılı Giriş Mesajı	27
Şekil 10: VoiceBot Ana Ekran.....	28
Şekil 11: Kullanicidata Tablosu.....	38
Şekil 12: Kullanicidata.db İçerisindeki Veriler	38

GENEL BİLGİLER

Adı ve Soyadı : Berfin Aleyna SUKUTİ
Bölümü : Bilgisayar Teknolojisi
Tez Danışmanı : Neşe Aslı KÖSE
Tez Türü ve Tarihi : Ön Lisans – Haziran 2023

ÖZET

GÖRME ENGELLİLER İÇİN SESLİ UYGULAMA

Dünya üzerinde engele sahip ve bu engeller yüzünden günlük yaşantılarına devam etmekte zorlanan pek çok insan vardır. Doğuştan ya da sonradan oluşan bu engellerin teknoloji ve bilim yardımı ile aşılabileceği ve hayatın çok daha fazla kolaylaştırılabileceği de bilinmektedir. Gelişen bilim ve teknolojinin ürünleri olarak pek çok uygulama mevcuttur. Görme engelleri bireyler için sesli, işitme engelli bireyler için görsel ve diğer engel unsurları için geliştirilen teknolojik araçların uygulamaları bu ürünlere örnektir.

Sesli uygulamaların en popüler örnekleri olan Siri ve Google Asistan, kullanıcıları için çok işlevseldir. Bu uygulamalardan ilham alınarak: sesli komut fonksiyonları, yol ve adres bulmak için harita araçları, mesaj ve e-posta gönderim işlemleri gibi çok fonksiyonlu uygulamalar geliştirilerek hayatı biraz daha kolaylaştırılabilir olacağının sonucuna varılmıştır.

Anahtar Kelimeler: Sesli uygulama, sesli asistan, sesi yazıya çevirmek, yazıyı sese çevirmek

GENERAL INFORMATION

Name and Surname : Berfin Aleyna SUKUTİ
Program : Computer Technologies
Supervisor : Neşe Aslı KÖSE
Degree Awarded and Date : Associate Degree – June 2023

ABSTRACT

AUDIO APPLICATION FOR THE VISUALLY HANDICAPPED

There are many people in the world who have disabilities and find it difficult to continue their daily lives because of these disabilities. It is also known that these congenital or acquired obstacles can be overcome with the help of technology and science, and life can be made much easier. There are many applications as products of developing science and technology. Examples of these products are the applications of technological tools developed for visually impaired individuals, visual and other disability elements for hearing impaired individuals.

Siri and Google Assistant, the most popular examples of voice apps, are very functional for their users. Inspired by these applications, it is concluded that life can be made a little easier by developing multi-functional applications such as voice command functions, map tools for finding directions and addresses, sending messages and e-mails.

Keywords: Voice application, voice assistance, text to voice, voice to text

1. GİRİŞ

Yapılan uygulama, özellikle işitme engelli bireylerin ihtiyaçlarını karşılamak ve hayatlarını kolaylaştırmak amacıyla yapılmıştır. VoiceBot, bir masaüstü uygulamasıdır. Sesleri yazıya dökerek, istenilen bilgiyi verme özelliği vardır.

1.1 VoiceBot Masaüstü Uygulama

Bu masaüstü uygulaması, görme ve/veya işitme engelli bireylerin de birbirleri ile iletişim kurabileceklerini hayal edilerek düşünüldü. Uygulamamızın adı: VoiceBot. Uygulamamızın içerisindeki özellikler arasında arama motoru ile arama yapma, saati söyleme, kayıt yapmak ve kayıtları göstermek bulunmaktadır. Uygulamaya giriş yapılırken öncelikle bizi bir kayıt sayfası açılacaktır. Eğer kayıt yapılmışsa buton yardımı ile giriş yapma sayfasına girebiliriz. Kayıt için istediğimiz bilgiler; isim, soy isim, parola, e-mail, adres, telefon ve cinsiyettir. Kayıt işlemin ardından giriş sayfasında ise kullanıcı mail adresi ve parola istenmektedir. Doğru bilgiler ile uygulamaya giriş yapıldıktan sonra karşımıza büyük bir yazı ekranı çıkacak ve yazdırma butonuna basıldığında sesli olarak komut söylenebilecektir. Girilen sesli komut ise yazı ekranında yazı şeklinde ifade edilecektir. Aynı şekilde yazıyı seslendirme butonu ile de girilen yazıyı sesli şekilde ifade edebiliriz.

VoiceBot uygulaması, görme veya işitme engelli bireylerin birbirleri ile daha rahat iletişim kurmaları hedefleyerek yapılmıştır. Kullanıcılarımızın hayatlarını kolaylaştırmak ve onların kendilerini daha detaylı ifade edebilecekleri oldukça basit bir arayüze sahip masaüstü uygulama platformu olarak hazırlanmıştır. Uygulama içerisinde kullanıcıların temel bilgilerini kaydederek sesli ve yazılı işlem yapmasını sağlamakla beraber, sayfa içerisinde çıkış yap butonu ile de uygulamadan çıkış yapabilirler.

1.2 VoiceBot Masaüstü Uygulama Avantajları

VoiceBot uygulamasının kullanıcıları, bilgisayarları nasıl ya da hangi özellikte olursa olsun bir sesli asistana sahip olmuş olacaklar. Yeni güncellemeleri ile beraber müşteri istek ve memnuniyetlerine göre yeni özellikler de eklenerek kullanıcılarımızın ihtiyaçlarını daha etkin şekilde giderebiliriz.

Bu sayede kullanıcılarımız daha çok iletişim kurabilir ve daha etkin bir iletişim sürecine başlayabilirler. Kendilerini ifade etmek ve bir diğer kişiye istenen kelimeleri aktarmak kullanıcılarımız için çok mutluluk verici olacağına ve öz güvenlerinin de çok daha fazla artacağını düşünüyoruz.

İşaret dilinin yanı sıra okuma yazma bilen bireyler için sesli asistan kullanımı, günlük yaşantılarında çok kolay şekilde olacağını ön görüyoruz.

1.3 VoiceBot Masaüstü Uygulamasını Destekleyen Platformlar

Windows işletim sistemine sahip bilgisayarı olan herkes Google Play veya eklenen başka platformlardan doğru yükleme işlemleri yapıldıktan sonra ulaşabilecektir.

1.4 Microsoft Windows İşletim Sistemi

Windows işletim sisteminin geliştiricisi Microsoft'tur. Kaynak türü sahipli yazılımdır. Yani kapalı kaynak kod kullanılmıştır. Windows, Microsoft'un geliştirmiş olduğu, dünya üzerinde en fazla kullanılan işletim sistemidir. İşletim sistemi ailesinden olan birinci Windows 1985'te satışa çıkarılmıştır.

Windows, NT çekirdek kullandığı için Unix işletim sistemine benzer bir yapıdadır. Microsoft çıkardığı ilk işletim sistemi olan MS-DOS ile tek yazılım kullanılabiliyordu. Ancak Windows işletim sistemi ile birden fazla yazılım ile çalışılabilir.

Masaüstü programlarının alanında en çok kullanılan işletim sistemi Windows'tur. Kendi alanındaki en büyük rakibi ise macOS'tur. Windows'un en son sürümü ise Windows 11 olmuştur. Windows 11, Ekim 2021 tarihinde piyasaya sunulmuştur. (1)

1.5 Windows İşletim Sisteminin Avantajları

Kullanım amacına göre işletim sistemleri avantajlıdır. Pazar genelinde en yaygın olan işletim sistemi ise Windows'tur. Grafiksel olarak kullanışlı bir arayüze sahiptir. Kullanıcılar tarafından basit ve anlaşılır şekilde yapılmış olması pazarda avantajlı duruma geçmesini sağlıyor. Görsel iletiler ile yazılım yürütme, komut verme vb. işlemleri mouse ve klavye yardımıyla girdi yapma kolaylığı sağlamaktadır.

Windows işletim sisteminin avantajları kısaca şu şekilde söylenebilir:

Bilgisayar kullanmakta zorlanan kişiler için görsel olarak kolay kullanım sağlayan çalışma ortamının oluşturulması.

Dosya isimlerinde uzun isimlere olanak sağlaması.

Çok fazla program desteğinin olması.

Girdi ve çıktı birimlerinde otomatik algılama sağlayan tak – çalıştır desteğinin olması. (2)

1.6 Google Play

Google Play, Android ve Windows işletim sistemleri için kullanılan uygulama indirme platformudur. Mobil Android cihazlarda uygulama indirme ve güncelleme işlemlerini yapar. Ancak Google hesabı ile Windows bilgisayarlardan da giriş yapılabilir. Masaüstü uygulamalar da bilgisayarlara Google Play ile indirilebilir ve güncellenebilir. Google Play aynı zamanda uygulamaları cihazdan kaldırma özelliğine de sahiptir.

1.7 Masaüstü Uygulama

Masaüstü uygulamalar, bilgisayarlar için yazılan yazılımlardır. Belirlenen bir programlama dili seçilerek, belirli bir hedef doğrultusunda yazılan bilgisayar programlarıdır. Bir bilgisayardaki temel yazılımlar dahil olmak üzere kullanıcılar için yapılan yazılımlar da masaüstü programlarındandır (Masaüstü Uygulamaları, 2023).

1.7.1 Masaüstü Uygulama Avantajları

Uygulamalar farklı işletim sistemleri ve farklı cihazlar arasında değişiklik göstermektedir. Her cihaza uygun olan programlar (uygulamalar) kendi aralarında avantajlıdır. Ancak masaüstü uygulamalar, bilgisayarlar için kullanılmaktadır. Bu uygulamalar işletim sistemlerinin çalışması amacıyla kullanılmasının yanı sıra yapılmış olan çeşitli otomasyon, oyun, eğitim, fotoğraf ve video düzenleme uygulamaları vb. tüm yazılımlar masaüstü uygulamalardır. Otomasyon uygulamaları işletmelerin veya kişilerin kayıt, güncelleme, silme gibi işlemleri kolayca yapmalarına olanak sağlar. Bilgisayar oyunları da oyuncular için bir oldukça avantajlı uygulamalardır. Fotoğraf ve video düzenleme uygulamaları da tasarlanan kolay arayüz kullanımı ile masaüstü uygulamalarına örnektir.

1.7.2 Masaüstü Uygulama Geliştirme

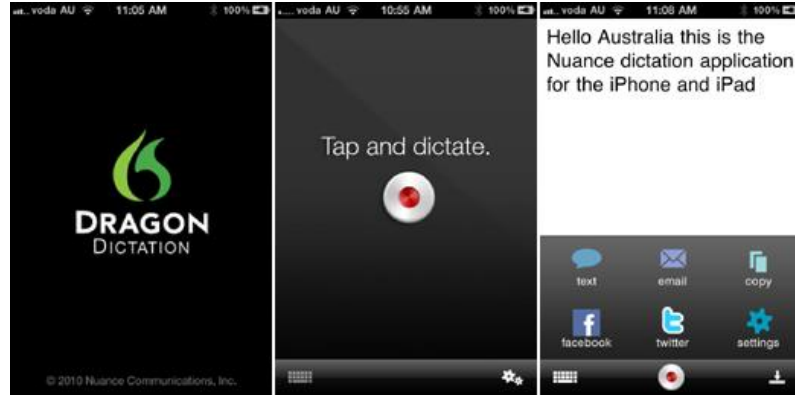
Masaüstü uygulama geliştirmek, bir problemi tanımlanmasından itibaren çözüme varıncaya kadar gelen süreç içerisinde ortaya çıkan üründür. Bir masaüstü uygulama geliştirmek demek, bilgisayar ortamında ihtiyaç olan alanda bir çözüm üretmektir. Uygulama geliştirmek için gerekli olan materyaller bilgisayar, ihtiyaç varsa internet, bir programlama dili, hedef programa uygun bir algoritma ve bir veri tabanıdır.

Geliştirilmek istenilen uygulamayı hayata geçirmek için öncelikle bir programlama dili seçilir, ardından algoritması kurulur ve bir ide yardımı ile program yazılır. Temel süreç belirtilen sıra içerisinde gerçekleştirilir.

1.8 Literatür Taraması

Literatür taraması; yapılan projenin örneklerini, bu örneklerin eksik yönlerini belirlemek ve projede bunların düzeltilmesi doğrultusunda araştırma yapmaktır. Yapılan proje, görme ve işitme engelli bireyler için hazırlanmıştır. Engelsiz bir iletişim hedefi ile yapılmış pek çok uygulama mevcuttur. Ancak uygulamalar mobil uygulamalardır. Android veya IOS işletim sistemlerinde kullanılan mobil yazılımlarıdır. Ben ise bir masaüstü programı hazırlayarak başka bir formatta bu uygulamayı hayata geçiriyorum.

1.8.1 Dragon Dictation: İşitme Engelliler İçin İletişim



Şekil 1: Dragon Dictation

Dragon Dictation, kullanıcının konuşmasını metne dönüştüren ve metin tabanlı işlemler yapabilmesini sağlayan bir ses tanıma uygulamasıdır. Dragon Dictation, mobil cihazlarda kullanılabilen popüler bir uygulamadır ve kullanıcılara konuşarak metin oluşturma, mesaj gönderme, e-posta yazma, sosyal medya paylaşımları yapma gibi işlemleri kolaylaştırır.

Uygulama, kullanıcının konuşmasını gerçek zamanlı olarak algılar ve bunu metne dönüştürür. Ses tanıma teknolojisi sayesinde kullanıcıların klavye kullanmadan yazı yazmalarını sağlar. Dragon Dictation, doğal dil işleme algoritmaları kullanarak kullanıcının sesini anlamlandırır ve metin olarak çıktı üretir.

Dragon Dictation, engelli bireyler için de kullanışlı bir uygulama olabilir. Örneğin, motor beceri kısıtlamaları olan bireyler için klavye kullanmak zor olabilirken, uygulama sayesinde konuşarak metin oluşturmak daha kolay olabilir. Ayrıca, görme engelli bireyler de uygulamanın metin çıktılarını sesli olarak dinleyerek kullanabilirler.

Dudak okumak veya işaret dili gibi özel becerilere sahip olmadan engelli bireylerle iletişim kurmak çok zor olabilir. Bu durumu çözüme ulaştırmak için ise metni sese veya sesi metne çevirmektir. Bu bağlamda Dragon Dictation uygulaması oldukça başarılıdır. Tek eksiği ise sadece Android cihazlarda hizmet vermesidir. Ayrıca uygulama İngilizce ve Türkçe de dahil olarak totalde 34 dil seçeneği vardır. Uygulama toplam 35.700 kullanıcı tarafından puanlanmış ve 4,5/5 puana sahiptir (TOPSAKAL, 2018).

1.8.2 JAWS Yazılım Uygulaması



Şekil 2: JAWS Yazılım

JAWS (Job Access With Speech), görme engelli bireyler için tasarlanmış bir ekran okuyucu yazılımdır. JAWS, bilgisayar ekranındaki metinleri sesli olarak okur ve kullanıcılara erişim sağlar. Bu yazılım, Windows işletim sistemi üzerinde çalışır ve bir dizi uygulama ve web tarayıcısıyla uyumludur.

JAWS, kullanıcılara metinleri, menüler, düğmeleri, bağlantıları ve diğer grafiksel öğeleri sesli olarak okuyarak bilgisayarın kullanılmasını kolaylaştırır. Ayrıca, klavye kısayolları ve komutlarıyla etkileşimli bir kullanım deneyimi sunar. Kullanıcılar, JAWS'ın konuşma hızını, ses tonunu ve diğer özelliklerini özelleştirebilir.

Bu yazılım, görme engelli bireylerin bilgisayarlarını kullanarak e-posta okuma, internet gezintisi, belge oluşturma, programlama, veritabanı erişimi gibi çeşitli görevleri gerçekleştirmelerini sağlar. Ayrıca, çeşitli dil ve lehçeleri destekler.

JAWS, engelli bireylerin bilgisayar erişimini kolaylaştırmak için önemli bir araçtır. Literatürde, JAWS gibi ekran okuyucu yazılımlarının kullanılabilirliği, etkinliği ve kullanıcı deneyimi üzerine yapılmış araştırmalar bulunabilir. Bu araştırmalar, yazılımın avantajları, kullanıcıların deneyimleri ve geliştirme alanları hakkında daha fazla bilgi sağlayabilir.

Bu tarz yazılımlar ile oluşturulan programlar, görme engelli bireylerin toplum ile daha uyumlu olmalarını ve herkes gibi eşit şartlar ile eğitimlerini görmeyi hak ediyorlar (İyigün & TORTOP, 2018).

1.8.3 CEYDA: Engelleri Aşan Uygulama



Şekil 3: CEYDA Sesli Asistan

Engelliler için teknolojik çözümler, hayatlarını kolaylaştırmak ve günlük yaşam aktivitelerinde bağımsızlık sağlamak açısından büyük bir öneme sahiptir. Bu kapsamda, Ceyda uygulaması, engelli bireylerin günlük hayatta karşılaştıkları zorlukları aşmalarına yardımcı olmayı hedefleyen bir mobil uygulamadır.

Ceyda, hem görme engelli bireyler hem de işitme engelli bireyler için kullanılabilir özelliklere sahiptir. Uygulama, kullanıcılara çevredeki nesneleri, insanları ve yol yönergelerini tanımlama konusunda yardımcı olur. Görme engelli bireyler için, Ceyda, nesneleri ve metinleri sesli olarak tanımlayabilir, yol tarifi sağlayabilir ve engeller hakkında uyarılar verebilir. İşitme engelli bireyler için ise, uygulama, sesleri metne dönüştürebilir ve iletişimde kullanıcıya yardımcı olmak için metin tabanlı bir arayüz sunabilir.

Ceyda'nın en önemli özelliklerinden biri, yapay zeka ve görüntü işleme teknolojilerinin entegrasyonudur. Bu sayede, uygulama çevredeki görüntüleri ve sesleri analiz edebilir, anlamlandırabilir ve kullanıcıya geri bildirimlerde bulunabilir. Uygulama ayrıca, kullanıcının tercihlerini ve önceden öğrenilen bilgileri dikkate alarak kişiselleştirilmiş bir deneyim sunabilir.

Ceyda'nın amacı, engelli bireylerin günlük yaşamda daha fazla bağımsızlık kazanmalarını sağlamaktır. Uygulama, sosyal etkileşimlerde, alışverişte, seyahatlerde ve diğer birçok durumda engelleri aşmayı destekleyerek yaşam kalitesini artırır. Ayrıca, Ceyda'nın gelişmeye devam eden bir platform olduğunu belirtmek önemlidir. Kullanıcı geri bildirimleri ve yeni teknolojik gelişmeler doğrultusunda sürekli olarak güncellenir ve iyileştirilir.

Sonu olarak, Ceyda uygulaması, engelli bireylerin gnlk hayatlarında karşılaştıkları zorlukları aşmalarını sağlamak amacıyla geliştirilmiş bir mobil uygulamadır. Teknoloji ve yapay zeka kullanımıyla, görme engelli ve işitme engelli bireylere destek sağlayarak daha bağımsız bir yaşam sürmelerine katkıda bulunur. Ceyda, engelleri aşma konusunda önemli bir adımdır ve gelecekte daha da gelişerek engelli bireylerin yaşamlarını kolaylaştırmaya devam edecektir (CEYD-A, 2023).

2. AMAÇ

Yapılan uygulamanın içerisinde, sesli olarak komut verme, sesli not kaydetme ve arama motorundan arama yapma faaliyetleri bulunmaktadır. Kullanıcılar, uygulamamıza kaydolduktan sonra ana sayfa içerisinde bir yazı alanı ve iki buton yardımı ile soru sorup cevap alabilirler.

Uygulamamıza herkes kayıt olabilir ve sonrasında hemen giriş yaparak kullanabilirler. Yaş veya cinsiyet sınırlaması olmaksızın kullanım açık bir masaüstü uygulamasıdır. Uygulamanın tek amacı günlük yaşantıları kolaylaştırmaktır.

3. GEREÇLER

Uygulamayı geliştirirken kullandığım gereçler Python dili, VSCode ve DB Browser for SQLite uygulamalarıdır. Uygulamayı oluştururken kullandığım gereçler:

- Uygulamanın oluşturulduğu ortam (IDE) Visual Studio Code,
- Kullandığım programlama dili Python,
- Uygulamaya dahil ettiğim veri tabanı ise SQLite DB Browser'dır.

3.1 Visual Studio Code Nedir



Şekil 4: Visual Studio Code

Visual Studio Code, pek çok programlama dilinin yazıldığı ortamdır. Programlama dillerinin IDE'si yani entegre geliştirme ortamıdır. Bu ortam içerisinde temel olarak JavaScript, TypeScript, Node.js dillerinin yanı sıra Python, C/C++, Java ve PHP dahil olmak üzere çok fazla dilde kullanılan bir entegre geliştirme ortamıdır.

Bilgisayarlar üzerinden IDE olarak kullanılabilir. Masaüstü veya dizüstü bilgisayar modelleri fark etmez ve uygulama geliştirmek için yeterlidir. Windows, MacOS ve Linux işletim sistemlerinde kullanılabilir. Çok güçlü geliştirme araçları vardır. Bu araçlar; kaynak kod düzenleyicisi, yazılan kodu tamamlama ve hata ayıklama araçlarıdır. Uygulama geliştiricileri için kolay bir arayüze ve işleve sahiptir. Çok fazla eklentisi vardır ve programların yazılımında çok fazla kolaylık sağlarlar. Bu eklentiler, programlama diline ait olan kütüphaneleri içererek gerekli kodların çalışmasını sağlar.

Visual Studio Code tamamen ücretsiz bir IDE'dir. Kaynak kodu düzenleyicisi olarak Microsoft tarafından geliştirilmiştir. Kodu yeniden yapılandırma gibi önemli geliştirici için yardımcı yöntemleri de bulunmaktadır. Uygulama geliştiricileri için çok önemli olan bir diğer özellik ise Git destekli olmasıdır. Git, bir sürüm denetim sistemidir. Geliştiriciler, projelerinin yerel bir kopyasını çıkartarak hem çevrimiçi hem de çevrimdışı hem de uzaktan çalışmayı olarak çalışmasına yardımcı olur. Bu sayede yeni sürümler eklendikten sonra önceki hali yedeklenmiş olarak Git'de durur. Bunun gibi pek çok özellik sayesinde en çok kullanılan IDE olarak Visual Studio Code geçmektedir (BULUT, 2020).

3.2 Python Nedir



Şekil 5: Python

Python üst düzey, dinamik, yorumlanabilir, nesne tabanlı ve yorumlanabilir bir programlama dilidir. İlk sürümü 1991 yılında çıkmıştır. Python, içerisinde pek çok kütüphane bulundurur ve bu doğrultuda da çok fazla kullanılan en kolay nesne odaklı programlama dilidir. Kolay ve sade tasarlanmış arayüzü sayesinde geliştiriciler çok rahat bir şekilde program geliştirebilirler.

Python ile yapılabilecekler şeyler; web uygulamaları geliştirmek, uygulama geliştirmek, veri bilimi ve makine öğrenmesi yapılabilir. Çok fazla alanda kullanılarak kendini geliştiren Python dili, içerisindeki kütüphaneler yardımı ile kaynak kodları tamamlama özelliği vardır. Python, tamamen ücretsiz bir dildir ve pek çok IDE' si vardır. Bu sayede pek çok platformda çalışarak hızlı geliştirmeler yapmaya olanak sağlar. Python' un kullanımları da çok çeşitlidir:

- a. Web geliştirme: Sunucu tarafında web geliştirme işlemi arka uç işlevlerini kapsamaktadır. Temelde web siteleri veri tabanı ile etkileşim içerisinde.
- b. Otomasyon: Python betik bir dildir. Böylece günlük işlemler otomatikleşerek programlanırlar. Bu betiklere örnek olarak temel matematik işlemlerini gerçekleştirme ve bir dosyayı başka türde bir dosyaya dönüştürme işlemleridir.
- c. Makine öğrenmesi / Veri bilimi: Veri bilimi, eldeki veri setinden istenilen bilgileri yakalamak, makine öğrenmesi de bilgisayara verilen verinin öğrenmesi ve tahminde bulunmasını sağlamaktır. Veri ayıklama ve veri etiketlemeler bu işlevlerdendir.

3.2.1 Python Dilinin Avantajları

Python, basit olması ve çok çeşitli kütüphaneleri ile çok fazla avantaja sahiptir. Bu nedenle de geliştiriciler tarafından çok fazla tercih edilmektedir. Python'un avantajları şu şekildedir:

- Python yapı olarak İngilizce söz dizimine çok benzediği için kaynak kodları okumak ve anlamak çok kolaydır.
- Uygulama geliştiriciler, Python ile istedikleri kodları daha az satırda yazarak tasarruf sağlayarak programlarının daha hızlı çalışmasını sağlayabilirler.
- Çok fazla kitaplığa sahip olduğundan pek çok görev ve istek için kullanılması uygun bir dildir.
- Python, diğer popülerliği fazla olan diller ile kolayca entegre olabilir.
- Python da çok popüler bir dil olduğundan kullanıcısı çok fazladır. Böylece hataları gidermek için çok fazla kaynağa sahip olmuş olursunuz.
- Python ile yazılan kodlar diğer işlerim sistemlerine de aktarılabilir.

3.2.2. Python Kütüphaneleri

Python dilinin özellikleri içerisinde çok fazla kütüphaneye sahip olduğunu anlatmıştım. Bu nedenle de çok popüler bir dildir ve avantajlıdır. En popüler kütüphaneler ise şunlardır:

- Matplotlib: Bilimsel olarak grafik oluşturmak ve taşımak için kullanılır.
- Pandas: Veri setindeki verileri yapılandırır ve optimize eder. Makine öğrenmesi alanında kullanılır.
- NumPy: Mantıksal şekil işleme ve dizi oluşturma gibi işlemleri gerçekleştirir.
- Request: Web geliştirmede kullanılan popüler bir kitaplıktır.
- OpenCV-Python: Görüntü işleme alanında kullanılır. Görüntü yakalama ve analiz yapabilir.
- Keras: Derin sinir ağı kütüphanesidir. Veri işleme ve görselleştirme işlemleri için kullanılır (Python Nedir?, 2023).

3.3 SQLite Nedir



Şekil 6: SQLite Veri Tabanı

SQLite, bir veri tabanı ve veri tabanı kütüphanesidir. Veri tabanı, verileri saklamak, güncellemek, eklemek, silmek gibi işlemlerin yapılabildiği bir platformdur. Veriler genellikle sunucularda tutulur ancak SQLite için bir sunucuya gerek yoktur. Ücretsizdir ve kullanımı çok kolaydır. Diğer veri tabanı uygulamaları gibi herhangi bir kurulum veya indirme gerekmez.

Kullanılması için bir kurulum gerekmiyor ancak oluşturulan ve güncellenen tablonuzu görmek isterseniz DB Browser for SQLite uygulamasını kurarak bu arayüz ile de tablo oluşturabilir, ekleme yapabilir, güncelleme yapabilir ve veri veya satır – sütun ya da tabloyu silebilirsiniz.

SQLite kullanılarak oluşturulan veri tabanını dosya halinde USB belleğe aktarabilir veya paylaşabilirsiniz. SQLite, dosya biçimli bir platformdur. Bu sebeple veri tabanı kopyalanıp başka cihazlarda da olduğu gibi kullanılabilir. Bu durum kullanıcıları için çok avantajlı bir durumdur.

SQLite, Python gibi dinamik diller ile beraber kullanılırsa çok daha pratik ve güvenilir olacaktır. SQLite boyut olarak da küçüktür ve kullanıcıların alanlarını kaplamaz. Bütün bu avantajları sayesinde de kısa sürede çok popülerleşmiş ve çok fazla kullanılmıştır. Farklı dil destekleri de mevcuttur (Çevik, 2022).

3.3.1 SQLite Yapısı ve Komutları

SQLite yapı olarak dosya halinde tutulmaktadır. Gerçek bir tablo üzerinden işlem yapmak yerine dosya üzerinden komutlar (sorgular) yazarak işlemleri gerçekleştirebiliriz.

Veri tabanı veriyi saklayarak sonradan sorgulara eklenebilecek değerleri içerisinde barındırır. Tablo benzeri sade ve kolay anlaşılır bir yapıda verileri saklar. Oluşturulmuş her bir tablonun farklı isimleri de vardır. Sorguları da tabloya verdiğimiz isim ile yazarız ve istediğimiz işlemler bu şekilde gerçekleşir.

Eğer kullandığınız işletim sistemi GNU/Linux ise SQLite Browser'ı kurabilirsiniz. Bunun için şöyle bir komut vardır: “sudo apt-get install sqlitebrowser”

Veri tabanını açmak istiyorsak sqlite programını çalıştırarak File / Folder > Open komutlarını takip etmeniz yeterlidir.

Python kullanarak bir tablo üzerinde işlem yapmak istiyorsak da öncelikle kütüphanemizi import etmemiz gerekmektedir. Örneğin: “import sqlite3 as sql” ile kitaplığımızı indiriyoruz. Artık veri tabanı ile bağlantı kurmak istediğimizde ise komutumuz şu şekildedir: “db = sqlite3.connect('veritabani_adi')”

Bağlantı kurulan connect metodu ile beraber imleç oluşturarak işlemleri veri tabanı tablomuz üzerinden gerçekleştirebiliriz. Bunun için gerekli komutlar:

“im = db.cursor()” ,

Tablo oluşturmak için: “im.execute(“””create table tablo_adi (a, b)”””)”

Veri eklemek için: “im.execute(“””insert into tablo_adi values (“x”,”y”)”””)”

İşlemlerin çalışması için ise “db.commit()” komutunu yazmak yeterlidir.

4. YÖNTEMLER

Projemi yaparken kullandığım araçların yanında bu araçlarda kullandığım yöntemler de bulunmaktadır. Bu yöntemler içerisinde projemde kullandığım tüm class, kütüphane ve methodlar dahildir.

Projeme başlarken kullandığım kütüphaneler şunlardır:

- a. Tkinter kütüphanesi: Bu kitaplık Python ile yapılan GUI uygulamalarında kullanılarak arayüz oluşturulmasını sağlar.
- b. Sqlite3 kütüphanesi: SQLite bir veri tabanıdır. Bu kütüphane kullanılarak tablo oluşturma ve veri ekleme, silme, güncelleme gibi işlemleri yapmayı sağlar.
- c. Speech_recognition kütüphanesi: Bu kütüphane yardımı ile sesli özellikler yazabiliyoruz. Bu kütüphane Google'a aittir ve bu sayede çok daha az kod yazarak sesli uygulama hazırlayabiliyoruz.
- d. Datetime kütüphanesi: Bu kütüphane ile bulunduğumuz zaman dilimini ve saati direkt olarak öğrenebiliyoruz.
- e. Webbrowser kütüphanesi: Bu kütüphane ile aramak istediğimiz herhangi bir konuyu varsayılan arama motoru üzerinden aratmamızı sağlıyor.
- f. Pyaudio kütüphanesi: Ses çalmak ve dinlemek için kullanılan kütüphanedir.
- g. Wave kütüphanesi: Wave, ses dosyalarını “.wav” formatında çalıştırmayı sağlar. Bu class içerisinde dosya sadece okunur veya sadece yazılır.
- h. Gtts kütüphanesi: Bu kütüphane ise yazıyı sese dönüştürmeye yarar.
- i. Pydub ve AudioSegment kütüphanesi: Ses segmenti modülüdür. Python için ses nesnelerinde sarmalayıcı olarak bilinir. Böylece her ses için ses listelerini döndürür.
- j. Play kütüphanesi: Bu kütüphane, .mp3 ses dosyalarını oynatmayı sağlayan modüldür.
- k. Os kütüphanesi: Bilgisayarımızdaki dosyaların okuma yazma işlemlerini gerçekleştirmeye olanak sağlayan modüldür.

Proje için yazmış olduğum tek class Kullanıcılar sınıfıdır. Bu sınıfı da arayüz içerisindeki alanlar ile veri tabanındaki bağlantıyı oluşturmak için kullandım.

```
class Kullanicilar:
```

```
    def __init__(self, isim=None, soyisim=None, parola=None, email=None,  
adres=None, telefon=None, cinsiyet=None):
```

```
        self.isim = isim
```

```
        self.soyisim = soyisim
```

```
        self.parola = parola
```

```
        self.email = email
```

```
        self.adres = adres
```

```
        self.telefon = telefon
```

```
        self.cinsiyet = cinsiyet
```

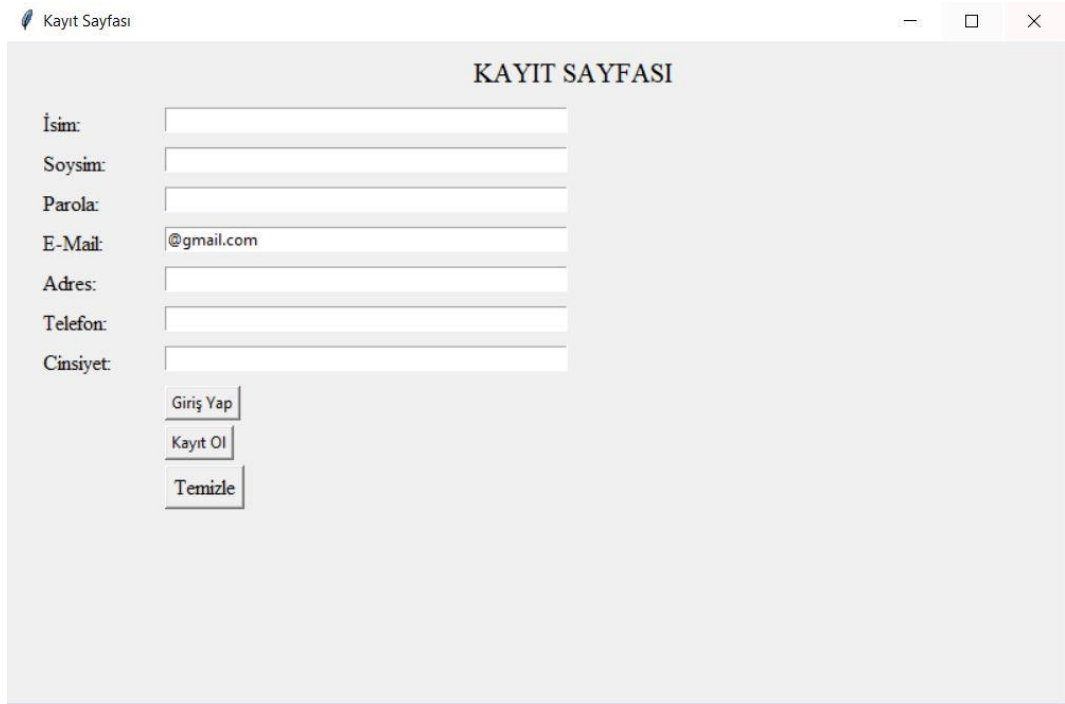
```
        self.conn = sqlite3.connect("KullaniciData.db")
```

```
        self.cursor = self.conn.cursor()
```

Proje için kullandığım metotlar ise şunlardır: (def)

- a. def create_table()
- b. def insert()
- c. def update_password()
- d. def delete_account()
- e. def delete_table()
- f. def giriş_window()
- g. def get_user_by_email()
- h. def chack_password()
- i. def giriş_onay()
- j. def sesi_yazma()
- k. def record()
- l. def response()
- m. def yaziyi_okuma()
- n. def kullanıcı_kayit()

4.1 Kayıt Ekranı (Başlangıç Ekranı) Arayüzü



Kayıt Sayfası

KAYIT SAYFASI

İsim:

Soyisim:

Parola:

E-Mail:

Adres:

Telefon:

Cinsiyet:

Giriş Yap

Kayıt Ol

Temizle

Şekil 7: VoiceBot Kayıt Ekranı

Kayıt ekranı programın başlangıç ekranıdır. Önce kayıt olup, sonradan “Giriş Yap” butonuna tıklayarak giriş yapma ekranına geçiş yapılabilir. Arayüz kodları ve fonksiyonlar şu şekildedir:

```
window = tk.Tk()

window.geometry("800x500")

window.title("Kayıt Sayfası")

def kullanıcı_kayit():

    isim = entry_isim.get()

    soyisim = entry_soyisim.get()

    parola = entry_parola.get()

    email = entry_email.get()

    adres = entry_adres.get()
```

```
telefon = entry_telefon.get()

cinsiyet = entry_cinsiyet.get()

insert(isim, soyisim, parola, email, adres, telefon, cinsiyet)

messagebox.showinfo("Başarılı", "Kayıt Başarılı")

kullanici = Kullanici()

kullanici.Kullanici(isim, soyisim, parola, email, adres, telefon, cinsiyet)

return
```

```
def temizle():
```

```
    entry_isim.delete(0, tk.END)

    entry_soyisim.delete(0, tk.END)

    entry_parola.delete(0, tk.END)

    entry_email.delete(0, tk.END)

    entry_adres.delete(0, tk.END)

    entry_telefon.delete(0, tk.END)

    entry_cinsiyet.delete(0, tk.END)
```

```
labelg = tk.Label(window, text = "KAYIT SAYFASI", font = "Times 16", fg = "black")
```

```
labelg.place(x = 350, y = 10)
```

```
labelisim = tk.Label(window, text = "İsim: ", font = "Times 12", fg = "black")
```

```
labelisim.place(x = 25, y = 50)
```

```
labelsoyisim = tk.Label(window, text = "Soyisim: ", font = "Times 12", fg = "black")
```

```
labelsoyisim.place(x = 25, y = 80)
```

```
labelparola = tk.Label(window, text = "Parola: ", font = "Times 12", fg = "black")
```

```
labelparola.place(x=25, y=110)
```

```
labelemail = tk.Label(window, text = "E-Mail: ", font = "Times 12", fg = "black")
```

```
labelemail.place(x=25, y=140)
```

```
labeladres = tk.Label(window, text = "Adres: ", font = "Times 12", fg = "black")
```

```
labeladres.place(x=25, y=170)
```

```
labeltel = tk.Label(window, text = "Telefon: ", font = "Times 12", fg = "black")
```

```
labeltel.place(x=25, y=200)
```

```
labelcinsiyet = tk.Label(window, text = "Cinsiyet: ", font = "Times 12", fg = "black")
```

```
labelcinsiyet.place(x=25, y=230)
```

```
entry_isim =tk.Entry(window, width=50)
```

```
entry_isim.insert(string="", index =0)
```

```
entry_isim.place(x = 120, y = 50)
```

```
entry_soyisim =tk.Entry(window, width=50)
```

```
entry_soyisim.insert(string="", index =0)
```

```
entry_soyisim.place(x = 120, y = 80)
```

```
entry_parola =tk.Entry(window, width=50, show="*")
```

```
entry_parola.insert(string="", index =0)
```

```
entry_parola.place(x = 120, y = 110)
```

```
entry_email =tk.Entry(window, width=50)
```

```
entry_email.insert(string="@gmail.com", index =0)
```

```
entry_email.place(x = 120, y = 140)
```

```
entry_adres =tk.Entry(window, width=50)
```

```
entry_adres.insert(string="", index =0)
```

```
entry_adres.place(x = 120, y = 170)
```

```
entry_telefon =tk.Entry(window, width=50)
```

```
entry_telefon.insert(string="", index =0)
```

```
entry_telefon.place(x = 120, y = 200)
```

```
entry_cinsiyet =tk.Entry(window, width=50)
```

```
entry_cinsiyet.insert(string="", index =0)
```

```
entry_cinsiyet.place(x = 120, y = 230)
```

```
giris = tk.Button(window, text = "Giriş Yap", fg = "black" , command=giriswindow)
```

```
giris.place(x = 120, y = 260)
```

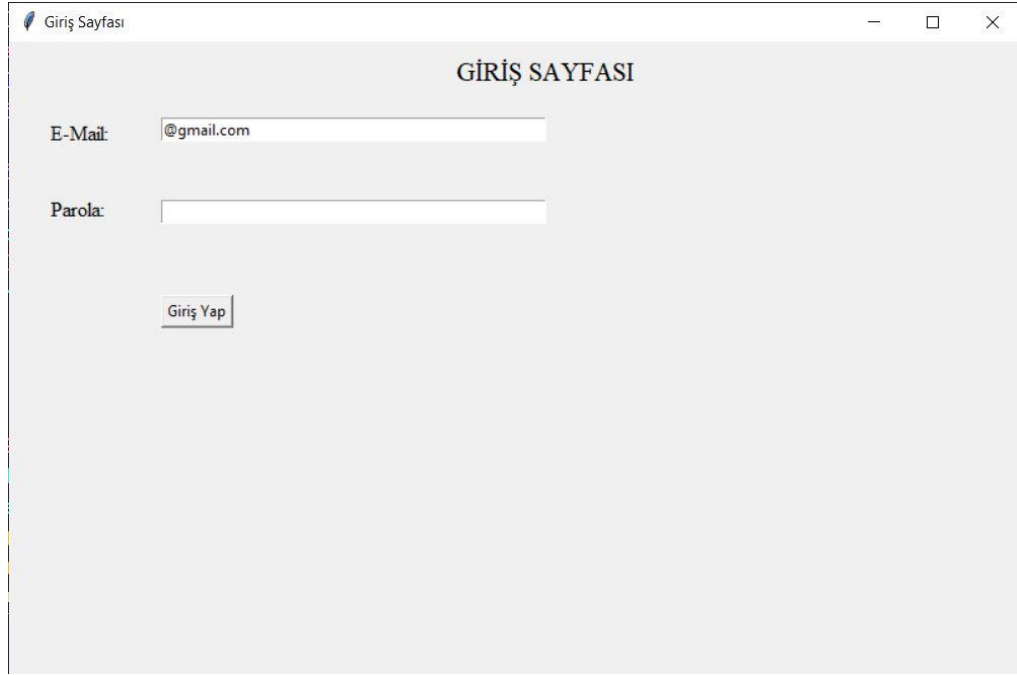
```
kayit = tk.Button(window, text = "Kayıt Ol", fg = "black", command =  
kullanici_kayit)
```

```
kayit.place(x = 120, y = 290)
```

```
button_temizle = tk.Button(window, text="Temizle", font="Times 12", fg="black",  
command = temizle)
```

```
button_temizle.place(x=120, y=320)
```

4.2 Giriş Ekranı Arayüzü



Şekil 8: VoiceBot Giriş Ekranı

Kayıt ekranında kayıt işlemlerimiz gerçekleştikten sonra giriş yapma ekranından email ve parola istenerek giriş sağlanıyor. Giriş arayüzü için kodlar ve metotlar:

```
def giriswindow():
```

```
    window.destroy()
```

```
    giriswindow = tk.Tk()
```

```
    giriswindow.geometry("800x500")
```

```
    giriswindow.title("Giriş Sayfası")
```

```
    labelg = tk.Label(giriswindow, text = "GİRİŞ SAYFASI", font = "Times 16", fg =  
"black")
```

```
    labelg.place(x = 350, y = 10)
```

```
labelemail = tk.Label(giriswindow, text = "E-Mail: ", font = "Times 12", fg =  
"black")
```

```
labelemail.place(x=30, y=60)
```

```
labelparola = tk.Label(giriswindow, text = "Parola: ", font = "Times 12", fg =  
"black")
```

```
labelparola.place(x=30, y=120)
```

```
entry_email =tk.Entry(giriswindow, width=50)
```

```
entry_email.insert(string="@gmail.com", index =0)
```

```
entry_email.place(x = 120, y = 60)
```

```
entry_parola =tk.Entry(giriswindow, width=50)
```

```
entry_parola.insert(string="", index =0)
```

```
entry_parola.place(x = 120, y = 125)
```

```
def get_user_by_email(email):
```

```
    conn = sql.connect('KullaniciData.db')
```

```
    cursor = conn.cursor()
```

```
    sel_command = """SELECT * FROM KullaniciData WHERE email = '{ }'"""
```

```
    cursor.execute(sel_command.format(email))
```

```
    user = cursor.fetchone()
```

```

        conn.commit()

        conn.close()

    return user

def check_password(email, parola):

    user = get_user_by_email(email)

    if user and user[3] == parola:

        return True

    else:

        return False

def giris_onay():

    email = entry_email.get()

    parola = entry_parola.get()

    if get_user_by_email(email):

        if check_password(email, parola):

            messagebox.showinfo("Giriş Başarılı", "Hoş Geldiniz!")

        else:

            messagebox.showerror("Hatalı Parola", "Girdiğiniz parola yanlış.")

    else:

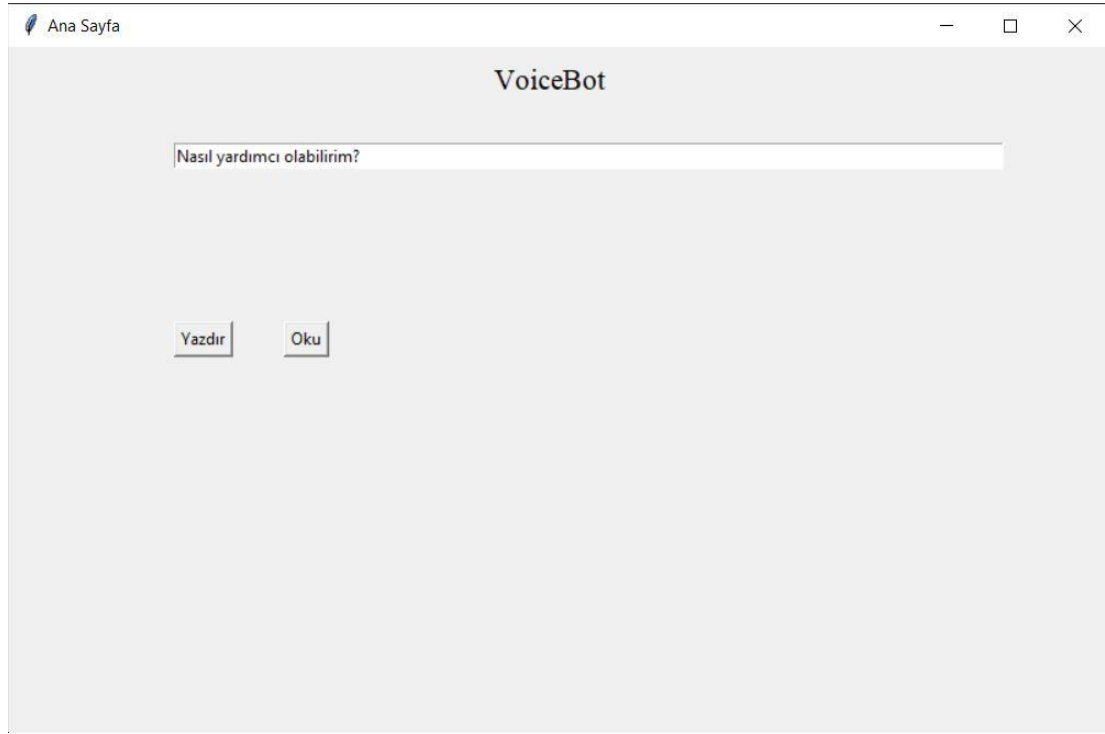
```

```
messagebox.showerror("Hatalı E-Mail", "Girdiğiniz e-mail adresi kayıtlı  
değil.")
```



Şekil 9: VoiceBot Başarılı Giriş Mesajı

4.3 Ana Ekran Arayüzü



Şekil 10: VoiceBot Ana Ekran

Giriş yapıldıktan sonra gelen giriş başarılı mesaj kutusundan sonra direkt olarak Ana Sayfa açılmaktadır. Yazdır butonuna basıldığında sesli komut vermemize ve söylediğimiz komuta göre program çalışır. Oku butonuna basıldığında ise önce sesli olarak okuma komutu veriyoruz ve ardından yazı ekranında Metni Giriniz: yazısı çıkıyor ve yazılan sesli olarak okunuyor. Bu işlemlerin kodları ve metotlar şu şekildedir:

```
mainwindow = tk.Tk()

mainwindow.geometry("800x500")

mainwindow.title("Ana Sayfa")

# window.destroy()

def sesi_yazma():

    # entry_veri = tk.Entry(mainwindow, width=100)
```

```

# entry_veri.insert(string="Nasıl yardımcı olabilirim?", index =0)

# entry_veri.place(x = 120, y = 70)

r = sr.Recognizer()

def record(ask = False):

    with sr.Microphone() as source:

        if ask:

            print(ask)

        audio = r.listen(source)

        voice = ""

        try:

            voice = r.recognize_google(audio, language = 'tr-TR')

        except sr.UnknownValueError:

            entry_veri.delete(0, tk.END)

            entry_veri.insert(tk.END, "Anlayamadım..")

        except sr.RequestError:

            entry_veri.delete(0, tk.END)

            entry_veri.insert(tk.END, "Sistem Çalışmıyor.")

        return voice

def response(voice):

    if "nasılsın" in voice:

        entry_veri.delete(0, tk.END)

```

```

entry_veri.insert(tk.END, "İyi senden?")

if "sesli kayıt yapmak istiyorum" in voice:

    # Kayıt ayarları

    CHUNK = 1024 # ses blokları

    FORMAT = pyaudio.paInt16 # ses formatı

    CHANNELS = 2 # ses kanalları

    RATE = 44100 # örnekleme hızı

    RECORD_SECONDS = 30 # kaydedilecek süre

    FILE_PATH = os.path.join("C:\kayitproje",
f"kayit_{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}.wav") #
kaydedilecek dosya yolu

    audio = pyaudio.PyAudio()

    # Ses kaydı

    stream = audio.open(format=FORMAT, channels=CHANNELS,
        rate=RATE, input=True,
        frames_per_buffer=CHUNK)

    #print("Kayıt başladı...")

    entry_veri.delete(0, tk.END)

    entry_veri.insert(tk.END, "Kayıt başladı...")

```

```

frames = []

for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):

    data = stream.read(CHUNK)

    frames.append(data)

print("Kayıt bitti.")

entry_veri.delete(0, tk.END)

entry_veri.insert(tk.END, "Kayıt bitti...")


# Ses dosyasına yazma

stream.stop_stream()

stream.close()

audio.terminate()


wf = wave.open(FILE_PATH, 'wb')

wf.setnchannels(CHANNELS)

wf.setsampwidth(audio.get_sample_size(FORMAT))

wf.setframerate(RATE)

wf.writeframes(b''.join(frames))

wf.close()


# if "oku" in voice:

#     def metni_sese_cevir(metin, dil='tr'):

```

```

#     tts = gTTS(text=metin, lang=dil)

#     ses_dosyasi = 'ses.mp3'

#     tts.save(ses_dosyasi)


#     ses = AudioSegment.from_mp3(ses_dosyasi)

#     play(ses)


#     #metin = input("Metni girin: ")

#     metin = entry_veri.get()

#     entry_veri.delete(0, tk.END)

#     entry_veri.insert(tk.END, "Metni girin: ")

#     entry_veri.delete(0, tk.END)

#     metni_sese_cevir(metin)

if "saat kaç" in voice:

    #print(datetime.now().strftime("%H:%M:%S"))

    entry_veri.delete(0, tk.END)

    entry_veri.insert(tk.END, "Saat: " + str(datetime.now().strftime("%H")))

if "arama yap" in voice:

    entry_veri.delete(0, tk.END)

    entry_veri.insert(tk.END, "Ne aramak istiyorsun?")

    search = record("ne aramak istiyorsun")

    url = "https://google.com/search?q="+search

    webbrowser.get().open(url)

    #print(search+ " için bulduklarım")

```

```

# if "mail gönder" in voice:

#   sender_email = "berfinsukuti1@gmail.com"

#   sender_password = "zfikowfprfohjkzx"

#   receiver_email = "berfinsukuti.01@hotmail.com"


#   msg = MIMEMultipart()

#   msg['From'] = sender_email

#   msg['To'] = receiver_email

#   msg['Subject'] = "Test Email from Python"

#   body = "Hello, this is a test email sent using Python!"

#   msg.attach(MIMEText(body, 'plain'))


#   smtp_server = smtplib.SMTP('smtp.gmail.com', 587)

#   smtp_server.starttls()

#   smtp_server.login(sender_email, sender_password)

#   text = msg.as_string()

#   smtp_server.sendmail(sender_email, receiver_email, text)

#   smtp_server.quit()

# print("Mail Gönderildi!")

if "tamamdır" in voice:

# print("görüşürüz")

exit()


# entry_veri = tk.Entry(mainwindow, width=100)

```

```

# entry_veri.insert(string="Nasıl yardımcı olabilirim?", index =0)

# entry_veri.place(x = 120, y = 70)


time.sleep(1)

while 1:

    voice=record()

    #print(voice)

    entry_veri.delete(0, tk.END)

    entry_veri.insert(tk.END, " " + voice)

    response(voice)


def yaziyi_okuma():

    r = sr.Recognizer()


def record(ask = False):

    with sr.Microphone() as source:

        if ask:

            print(ask)

            audio = r.listen(source)

            voice = ""

            try:

                voice = r.recognize_google(audio, language = 'tr-TR')

```

```

except sr.UnknownValueError:

    #print("anlayamadım.")

    labelValueError = tk.Label(mainwindow, text="Anlayamadım.", font
= "Times 12", fg = "black")

    labelValueError.place(x = 100, y = 250)

except sr.RequestError:

    #print("sistem çalışmıyor.")

    labelReqError = tk.Label(mainwindow, text="Sistem Çalışmıyor...",
font = "Times 12", fg = "black")

    labelReqError.place(x = 100, y = 280)

return voice

def response(voice):

    entry_veri.delete(0, tk.END)

    entry_veri.insert(tk.END, "Metni girin: ")

    if "oku" in voice:

        metin = entry_veri.get()

        entry_veri.delete(0, tk.END)

        entry_veri.insert(tk.END, "")

        def metni_sese_cevir(metin, dil='tr'):

            tts = gTTS(text=metin, lang=dil)

            ses_dosyasi = 'ses.mp3'

            tts.save(ses_dosyasi)

```

```

        ses = AudioSegment.from_wav(ses_dosyasi)

        play(ses)

        #metin = input("Metni girin: ")

        metni_sese_cevir(metin)

        time.sleep(1)

        while 1:

            voice=record()

            #print(voice)

            entry_veri.delete(0, tk.END)

            entry_veri.insert(tk.END, " " + voice)

            response(voice)

        label = tk.Label(mainwindow, text = "VoiceBot", font = "Times 16", fg =
"black")

        label.place(x = 350, y = 10)

        entry_veri = tk.Entry(mainwindow, width=100)

        entry_veri.insert(string="Nasıl yardımcı olabilirim?", index =0)

        entry_veri.place(x = 120, y = 70)

        yazdirma = tk.Button(mainwindow, text="Yazdır", fg = "black",
command=sesi_yazma)

        yazdirma.place(x=120, y=200)

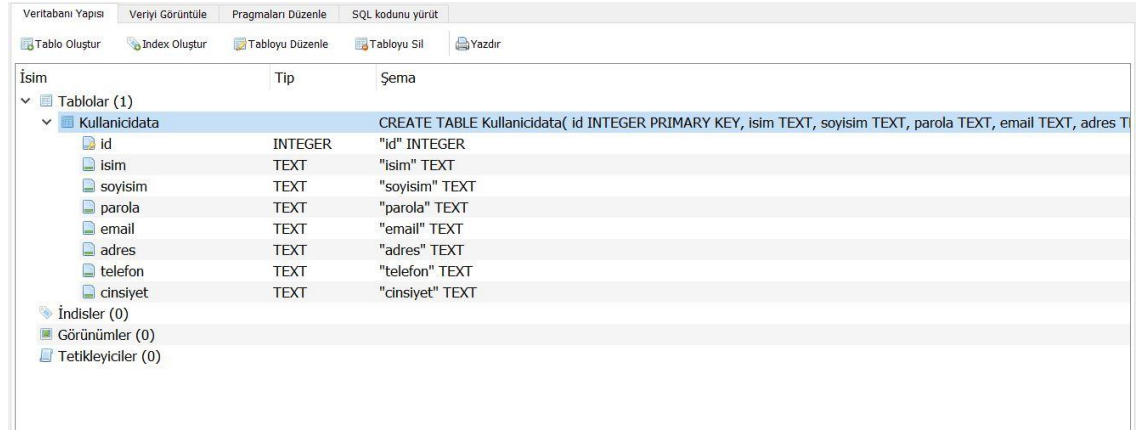
```

```
okuma = tk.Button(mainwindow, text="Oku", fg = "black",  
command=yaziyi_okuma)
```

```
okuma.place(x = 200, y = 200)
```

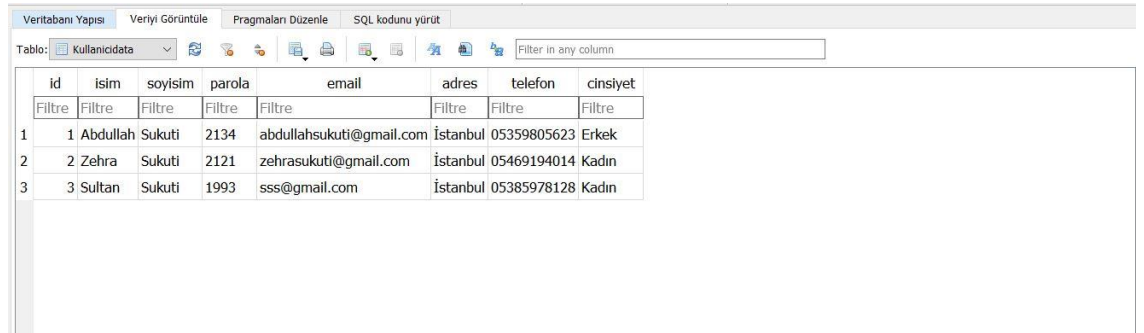
4.3 Uygulamamın Veri Tabanı Kullanıcılar.db

Kullanıcıdata.db veri tabanı tablosunun SQLite DB Browser görüntüleri şöyledir:



İsim	Tip	Şema
Tablolar (1)		
▼ Kullanıcıdata		CREATE TABLE Kullanıcıdata(id INTEGER PRIMARY KEY, isim TEXT, soyisim TEXT, parola TEXT, email TEXT, adres T
id	INTEGER	"id" INTEGER
isim	TEXT	"isim" TEXT
soyisim	TEXT	"soyisim" TEXT
parola	TEXT	"parola" TEXT
email	TEXT	"email" TEXT
adres	TEXT	"adres" TEXT
telefon	TEXT	"telefon" TEXT
cinsiyet	TEXT	"cinsiyet" TEXT
İndisler (0)		
Görünümler (0)		
Tetikleyiciler (0)		

Şekil 11: Kullanıcıdata Tablosu



	id	isim	soyisim	parola	email	adres	telefon	cinsiyet
1	1	Abdullah	Sukuti	2134	abdullahsukuti@gmail.com	İstanbul	05359805623	Erkek
2	2	Zehra	Sukuti	2121	zehrasukuti@gmail.com	İstanbul	05469194014	Kadın
3	3	Sultan	Sukuti	1993	sss@gmail.com	İstanbul	05385978128	Kadın

Şekil 12: Kullanıcıdata.db İçerisindeki Veriler

Uygulama için kullandığım veri tabanı SQLite' tır. Bu tabloyu ve içerisindeki verileri DB Browser üzerinden değil, projemdeki Python kodu içerisinde yazarak oluşturdum ve işlem yaptırđım.

4.4 Tablo Oluşturma Kodları

```
import sqlite3 as sql

import sqlite3

def create_table():

    conn = sql.connect("KullaniciData.db")

    cursor = conn.cursor()

    cursor.execute("""CREATE TABLE IF NOT EXISTS KullaniciData(

        id INTEGER PRIMARY KEY,

        isim TEXT,

        soyisim TEXT,

        parola TEXT,

        email TEXT,

        adres TEXT,

        telefon TEXT,

        cinsiyet TEXT

    )""")

    conn.commit()

    conn.close()
```

4.5 Tabloya Veri Ekleme Kodları

```
import sqlite3 as sql
```

```
import sqlite3
```

```
def insert(isim, soyisim, parola, email, adres, telefon, cinsiyet):
```

```
    conn = sqlite3.connect("KullaniciData.db")
```

```
    cursor = conn.cursor()
```

```
    add_command = """INSERT INTO KullaniciData(isim, soyisim, parola, email,
adres, telefon, cinsiyet) values(?, ?, ?, ?, ?, ?, ?)"""
```

```
    #data = (entry_isim, entry_soyisim, entry_parola, entry_email, entry_adres,
entry_telefon, entry_cinsiyet)
```

```
    cursor.execute(add_command, (isim, soyisim, parola, email, adres, telefon,
cinsiyet))
```

```
    conn.commit()
```

```
    conn.close()
```

4.6 Parola Değiştirme Kodu

```
import sqlite3 as sql
```

```
import sqlite3
```

```
def update_password(email, newParola):
```

```
    conn = sql.connect('KullaniciData.db')
```

```
    cursor = conn.cursor()
```

```
    upd_command = """UPDATE KullaniciData SET parola = '{}' WHERE email =  
    '{}'"""
```

```
    cursor.execute(upd_command.format(newParola, email))
```

```
    conn.commit()
```

```
    conn.close()
```

4.7 Uygulamadan Hesap Silme Kodu

```
import sqlite3 as sql

import sqlite3

def delete_account(email):

    conn = sql.connect('KullaniciData.db')

    cursor = conn.cursor()

    dlt_command = """DELETE FROM KullaniciData WHERE email = '{}'"""

    cursor.execute(dlt_command.format(email))

    conn.commit()

    conn.close()
```

4.8 Tablo Silme Kodu

```
import sqlite3 as sql

import sqlite3

def delete_table():

    conn = sql.connect('KullaniciData.db')

    cursor = conn.cursor()

    cursor.execute("""DROP TABLE KullaniciData""")

    conn.commit()

    conn.close()
```

5. SONUÇ

Bu uygulamanın yapılmasındaki hedef, özel bireylerin de günlük hayatlarında iletişim kurabileceği, günlük ve temel olan bazı şeyleri sesli bir şekilde gerçekleştirmektir. Bu hedef doğrultusunda bu uygulamanın ihtiyaçlar doğrultusunda her zaman güncel tutularak herkese ulaşmasını hedefliyorum.

Bir şeyi sesli olarak dile getirememek ya da sürekli sesle bir şeyler not tutmak istemek çok doğal isteklerdir. Ancak bunu yapabilmek için normal bir cihaz kullanmak yeterli olmayabilir. Özellikle mobil alanda yapılmış olan uygulama örneklerinin bir masaüstü uygulama olarak da kullanılabilmesi, özel insanlar için çok kullanışlı olacaktır.

Gelişmenin ve yeniliğin her zaman bizleri ayakta tutacağına inanıyorum ve bu yüzden de bu uygulamayı sürekli geliştirmek ve güncel tutmak istiyorum. Sadece masaüstü uygulaması olarak değil bir de mobil uygulama olarak kullanıcılara sunulması da hedefler arasındadır.

Kullanıcılarımızın günlük yaşantılarına dahil olarak onlarla beraber daha kolay günler geçirmemizi umuyorum.

6. TARTIŞMA

Uygulamanın mobil cihazlar üzerinden örnekleri ağırlıklıdır. Bu nedenle bu uygulamanın daha çok kişiye yarar sağlaması için mobil sürümünün de yapılması gerekmektedir.

Sesli komutlar günlük yaşamı kolaylaştırmaya yetmeyebilir, bu nedenle ilerleyen zamanlarda toplu kamu alanlarının ve afet toplanma alanlarına yönlendiren bir uygulama olması da deprem ve diğer afet bölgesinde olduğumuz gerçeğini hatırlatarak daha yararlı bir uygulamaya dönüşeceğini düşünüyorum.

KAYNAKÇA

- BULUT, C. (2020). *Visual Studio Code Nedir?* argenova: <https://www.argenova.com.tr/visual-studio-code-nedir> adresinden alındı
- CEYD-A. (2023, 01 09). *Geliştirilebilir Sanal Asistan Platformu*. Ceyd-A Blog: <https://blog.ceyd-a.com/> adresinden alındı
- Çevik, A. (2022). *SQLite Nedir?* Alican Çevik: <http://blog.alicancevik.com/sqlite-nedir/> adresinden alındı
- İyigün, S. Ç., & TORTOP, H. S. (2018, 07 26). Özel Eğitimde Yenilikçi Uygulamalar Görme Engelli Bireyler İçin İnovatif ve Yenilikçi Teknolojik Araç Tasarımları ve Yaşam Doyumlarına Etkisi . İstanbul, Türkiye.
- Masaüstü Uygulamaları*. (2023). Webtures: <https://www.webtures.com/tr/blog/sozluk/masaustu-uygulamalari/> adresinden alındı
- Python Nedir?* (2023). AWS Amazon: <https://aws.amazon.com/tr/what-is/python/#:~:text=Python%3B%20web%20uygulamalar%C4%B1%2C%20yaz%C4%B1l%C4%B1m%20geli%C5%9Firme,%C3%A7al%C4%B1%C5%9Ft%C4%B1r%C4%B1labildi%C4%9Fi%20i%C3%A7in%20Python'%C4%B1%20kullan%C4%B1r.> adresinden alındı
- TOPSAKAL, D. Y. (2018, 08 04). AKILLI TURİZM KAPSAMINDA ENGELLİ DOSTU MOBİL HİZMETLER: TÜRKİYE TURİZM İÇİN ÖNERİLER. Adana, Türkiye.

ÖZGEÇMİŞ

Berfin Aleyna SUKUTİ

16 Haziran 2001 tarihinde İstanbul’da doğdu. İlk ve orta öğrenimini İsmail Hakkı Uludağ İlköğretim okulunda okudu. Sonrasında ise lise eğitimini Haydar Akın Mesleki ve Teknik Anadolu Lisesi’nde, İstanbul’da tamamladı. Burada Veri Tabanı Programcılığı alanında eğitimini tamamlayarak 2019 yılında yaz stajını tamamladı. 2021 yılında ise Haliç Üniversitesi’nde Bilgisayar Teknolojisi bölümüne giriş yaptı. 2023 yılında lisans eğitimine devam etmek üzere Haliç Üniversitesi’nden mezun oldu.