**T.C.**

**KIRKLARELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**



**YAZ16408 BULUT BİLİŞİM**

**Bütünleme Ödevi**

**GRUP ÜYELERİ**

**A person wearing a black shirt

Description automatically generated**

**1180505612 – Mekan MYRADOV**

**PROJENİN GİTHUB LİNKİ**

<https://github.com/MekanMyradov/aws_polly>

**KIRKLARELİ, 2022**

İçindekiler

[**1** **Giriş** 3](#_Toc107089913)

[**2** **AWS ile ilgili İşlemler** 3](#_Toc107089914)

[**3** **AWS Kullanıcısını Boto3’e Bağlama** 3](#_Toc107089915)

[**4** **Metni Konuşmaya Çeviren Uygulama** 4](#_Toc107089916)

# **Giriş**

Amazon Polly, metni konuşmaya çeviren bulut tabanlı bir sistemdir. Amazon Polly birçok dili ve gerçekçi sesleri desteklemektedir. Bu sistemi kullanırken sadece metinden sentezlenen konuşma ödeme yapılır. Sentezlenen konuşmalar ileride kullanılmak üzere kaydedilebilir ve ücretsiz olarak kullanılabilir.  
Bu çalışmada Python Boto3 kütüphanesini kullanarak metni konuşmaya çeviren bir uygulama yapılmıştır.

# **AWS ile ilgili İşlemler**

Geliştirilen uygulamanın AWS’e erişebilmesi için **tmp\_user** isimli Yönetici yetkilerine sahip bir kullanıcı oluşturulmuştur. Bu kullanıcıyı **Şekil 1**'de görebilirsiniz.

Graphical user interface, text, application

Description automatically generated

**Şekil 1:** **tmp\_user** isimli kullanıcı

# **AWS Kullanıcısını Boto3’e Bağlama**

AWS'ye Python Boto3 kullanarak erişmek için Boto3'ün hangi kullanıcıdan erişeceğini belirtmek gerekir. Bu, **awscli** adlı Python kütüphanesi kullanılarak yapılabilir. Bu kütüphane **pip install awscli** komutu ile kurulabilir.

Yukarıda oluşturulan **tmp\_user** isimli kullanıcıyı Boto3'e bağlamak için **Şekil 2**'deki gibi bir profil oluşturulur.

Text

Description automatically generated

**Şekil 2:** Oluşturulan profil

Kullanıcı Boto3'e bağlandıktan sonra yukarıdaki **aws** komutu ile oluşturulan profilin adı ve AWS hizmeti belirtilerek AWS'nin tüm hizmetlerine kod aracılığıyla erişilebilir.

# **Metni Konuşmaya Çeviren Uygulama**

Gerekli kütüphaneler kod dosyasına dahil edilir:

import tkinter as tk

import boto3

import os

import sys

from tempfile import gettempdir

from contextlib import closing

720x480 boyutunda pencere oluşturulur ve bu pencerenin başlık kısmına Polly yazılır:

window = tk.Tk()

window.geometry("720x480")

window.title("Polly");

Oluşturulan pencerenin içinde 10 satırlık bir metin alanı oluşturulur ve bu metin alanı çerçevelenir:

text\_field = tk.Text(window, height=10)

text\_field.pack()

Amazon Polly'ye **tmp\_user** isimli kullanıcıdan bağlanarak, metin alanına yazılan cümleyi konuşmaya çeviren ve o anki dizinde geçici bir klasör açarak bu konuşmayı kaydeden ve kaydedilen konuşmayı bilgisayarın varsayılan medya oynatıcısıyla oynatan fonksiyon implemente edilir:

def get\_text():

    # Connect to the AWS

    aws\_connect = boto3.session.Session(profile\_name="tmp\_user")

    client = aws\_connect.client(service\_name="polly", region\_name='us-east-1')

    text = text\_field.get("1.0", "end")

    # print(text)

    speech = client.synthesize\_speech(Text=text, Engine='neural', OutputFormat='mp3',

VoiceId='Joanna')

    # print(speech)

    # Check if there is the audio in a returned result. If there is save the audio to a file. Otherwise raise exception.

    if "AudioStream" in speech:

        with closing(speech['AudioStream']) as stream:

            audio\_path = os.path.join(gettempdir(), "speech.mp3")

            try:

                with open(audio\_path, "wb") as file:

                    file.write(stream.read())

            except IOError as error:

                print(error)

                sys.exit(-1)

    else:

        print("Could not found AudioStream")

        sys.exit(-1)

    if sys.platform == "win32":

        os.startfile(audio\_path)

Pencerenin içinde 1 satır kalınlığında ve 10 sütun genişliğinde bir buton oluşturulur ve bu butonun içine “Read Aloud” yazısı yazılır. Oluşturulan butona tıklandığında yukarıda gösterilen **get\_text()** fonksiyonu çağrılır.

button = tk.Button(window, height=1, width=10, text="Read Aloud", command=get\_text)

button.pack()

Aşağıdaki kod parçası, pencerenin manuel olarak kapatılana kadar kalmasını sağlar:

window.mainloop()

Yukarıda verilen kod parçaları birleştirilip çalıştırıldığında uygulamanın sonucu **Şekil 3**’teki gibi olacaktır.  
“Read Aloud” butonuna tıklandığında ise bilgisayarın varsayılan medya oynatıcısı yazılan metini konuşmaya çevirecektir.

Graphical user interface, application

Description automatically generated

**Şekil 3:** Uygulamanın sonucu