DAYANANDA SAGAR COLLEGE OF ENGINEERING

Department of Electronics and Communication Engineering

Assignment

Program: B.E.   **Semester:** 6
Course: Python Programming   **Section:** D
Course Code: 19CE6IEPYP   **Date:** 28-06-22

A Report on:

"Automation of the Text to Speech and Speech to Text use cases using OSS libraries"

Submitted by:

1DS19EC704   Amrit Shandilya
1DS19EC715   Ekta Sinha
1DS19EC725   Narayan Kumar

Faculty In-charge:

Dr. S Thenmozhi

Associate Professor
Dept. of Electronics and Communication Engineering

Dayananda Sagar College of Engineering,
Shavige Malleshwara Hills,Kumarswamy Layout,Bengaluru -560078

2022-23

Signature of Faculty In-charge

USE CASE:2

Automation of the Text to Speech and Speech to Text use cases using OSS libraries

Use Case Description:

Week	Task	Input	Processing Step	Output
1	Speech recognization	Audio via mic	identify a open source speech recognization library and understand the functionalities	Text of Audio
2	Text recognization	Text	identify a open source text recognization library and understand the functionalities	Audio of text
3	Validation	Audio and text	Compare audio and text, improve if needed	Audio and text should match

Section: D

Team Members:
Amrit Shandilya
Ekta Sinha
Narayan Kumar

Algorithm/Flowchart:
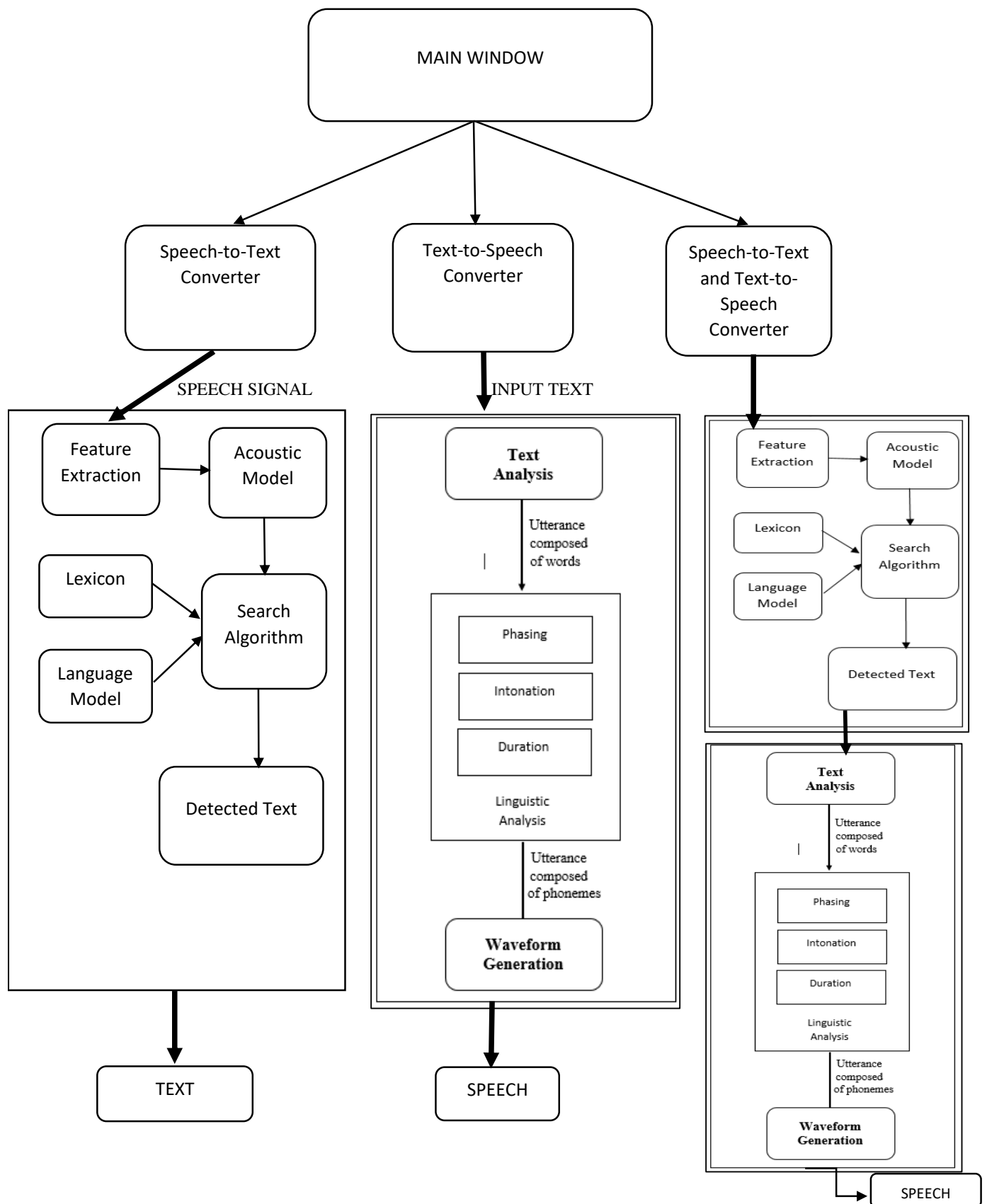
Program (with Comments):

- ### Python Program with GUI:

```python
1.  #importing all the necessary libraries
2.  from tkinter import *  #importing tkinter for gui
3.  from requests import delete
4.  import speech_recognition as sr  #importing speech_recognition for  speech recognition
5.  import pyttsx3  #importing pyttsx3 for text to speech conversion
6.
7.  e=pyttsx3.init()  #creating an object of pyttsx3
8.
9.  mainwindow= Tk()  #creating the main page for the gui
10. #defining the features of main page
11. mainwindow.title(' Text-To-Speech and Speech-To-Text Converter')
12. mainwindow.geometry('500x500')
13. mainwindow.configure(bg='light yellow')
14.
15. def listen_and_recognise():  #defining the function to listen and recognise the speech
    input
16.     while True:
17.         r = sr.Recognizer()  #creating an object of speech recognition recognizer class
18.         with sr.Microphone() as s:  #taking microphone as input source
19.             r.adjust_for_ambient_noise(s, duration=0.2)
20.             a=r.listen(s)
21.             try:
22.                 text = r.recognize_google(a,language="en-IN")  #recognise speech from
    the audio according to the specified dialect
23.             except sr.UnknownValueError:
24.                 print("Google Speech Recognition could not understand audio")
25.             except sr.RequestError as e:
26.                 print("Could not request results from Google Speech Recognition
    service")
27.         return text  #return text output
28.
29. def listen_recognise_and_speak():  #defining a function for recognising the speech and
    returning speech as well as text
30.     e=pyttsx3.init()  #creating an object of pyttsx init class
31.     r = sr.Recognizer()
32.     with sr.Microphone() as s:
33.         r.adjust_for_ambient_noise(s, duration=0.2)
34.         a = r.listen(s)
35.         text = r.recognize_google(a)
36.     e.say(text) #returns speech output
37.     e.runAndWait()
38.     return text #returns text output
39.
40. def speak(message): #defining a function to speak the output message
41.     e.say(message)  #speech output
42.     e.setProperty("rate", 178)
43.     e.runAndWait()
44.
45. def from_text_to_Speech():   #defining function for text to speech conversion
46.     #specifying the features of text to speech conversion window
47.     texttospeechwindow = Toplevel(mainwindow)
48.     texttospeechwindow.title('Text-to-Speech Converter ')
49.     texttospeechwindow.geometry("500x500")
50.     texttospeechwindow.configure(bg='snow2')
51.
52.     Label(texttospeechwindow, text='Text-to-Speech Converter ', font=("Comic Sans MS",
    18), bg='beige').place(x=90 , y=10)
53.
54.     text = Text(texttospeechwindow, height=10, width=50, font=12)
55.     text.place(x=22, y=60)
56.
```

```python
57.    speakbutton = Button(texttospeechwindow, text='Listen', bg='coral', command=lambda:
    speak(str(text.get(1.0, END))))
58.    speakbutton.place(x=200, y=250)  #customizing the listen button
59.
60.    clearbutton = Button(texttospeechwindow, text='Clear', bg='coral', command=lambda:
    str(text.delete("1.0","end")))
61.    clearbutton.place(x=250, y=250)  #customizing the clear button
62.
63. def from_speech_to_text():  #defining function for speech to text Conversion
64.    #specifying the features of speech to textconversuon window
65.    speechtotextwindow = Toplevel(mainwindow)  #used to crate the main window for the
    sub-window
66.    speechtotextwindow.title('Speech-to-Text Converter ')
67.    speechtotextwindow.geometry("500x500")
68.    speechtotextwindow.configure(bg='pink')
69.
70.    Label(speechtotextwindow, text='Speech-to-Text Converter ', font=("Comic Sans MS",
    18), bg='Lavender').place(x=90 , y=10)
71.
72.    text = Text(speechtotextwindow, font=12, height=10, width=50)
73.    text.place(x=22, y=100)
74.
75.    recordbutton = Button(speechtotextwindow, text='Record', bg='coral', command=lambda:
    text.insert(END, listen_and_recognise()))
76.    recordbutton.place(x=200, y=60) #customizing the record button
77.
78.    clearbutton = Button(speechtotextwindow, text='Clear', bg='coral', command=lambda:
    text.delete("1.0","end"))
79.    clearbutton.place(x=260, y=60) #customizing the clear button
80.
81. def both():  #defining a function for both speech to text and text to speech Conversion
82.    #specifying the specifications of the speech to text and text to speech window
83.    speechtotextandTextTOSpeechwindow = Toplevel(mainwindow)
84.    speechtotextandTextTOSpeechwindow.title('Speech-to-Text and Text-to-Speech Converter
    ')
85.    speechtotextandTextTOSpeechwindow.geometry("500x500")
86.    speechtotextandTextTOSpeechwindow.configure(bg='khaki')
87.
88.    Label(speechtotextandTextTOSpeechwindow, text='Speech-to-Text and Text-to-Speech
    Converter ', font=("Comic Sans MS", 16), bg='#9A7B4F').place(x=15 , y=10)
89.
90.    text = Text(speechtotextandTextTOSpeechwindow, font=12, height=10, width=50 )
91.    text.place(x=22, y=100)
92.
93.    recordbutton = Button(speechtotextandTextTOSpeechwindow, text='Record', bg='coral',
    command=lambda: text.insert(END, listen_recognise_and_speak()))
94.    recordbutton.place(x=200, y=60)  #customizing the record button
95.    clearbutton = Button(speechtotextandTextTOSpeechwindow, text='Clear', bg='coral',
    command=lambda: text.delete("1.0","end"))
96.    clearbutton.place(x=260, y=60)   #customizing the clear button
97.
98. #customizing main window
99. Label(mainwindow, text=' Text-To-Speech and Speech-To-Text Converter',
100.      font=('Comic Sans MS', 20), bg='light salmon', wrap=True,
    wraplength=450).place(x=75, y=20)
101.  texttospeechbutton = Button(mainwindow, text='Text-To-Speech Conversion', font=('Times
    New Roman', 16), bg='light slate blue', command=from_text_to_Speech)
102.  texttospeechbutton.place(x=120, y=250)
103.  speechtotextbutton = Button(mainwindow, text='Speech-To-Text Conversion', font=('Times
    New Roman', 16), bg='light slate blue', command=from_speech_to_text)
104.  speechtotextbutton.place(x=120, y=150)
105.  speechtotextandTextTOSpeechbutton = Button(mainwindow, text='Speech-to-Text and Text-
    to-Speech Converter ', font=('Times New Roman', 16), bg='light slate blue',
    command=both)
106.  speechtotextandTextTOSpeechbutton.place(x=50, y=350)
107.  mainwindow.update()  #shows the updated screen
108.  mainwindow.mainloop() #infinite loop used to run the application window
```
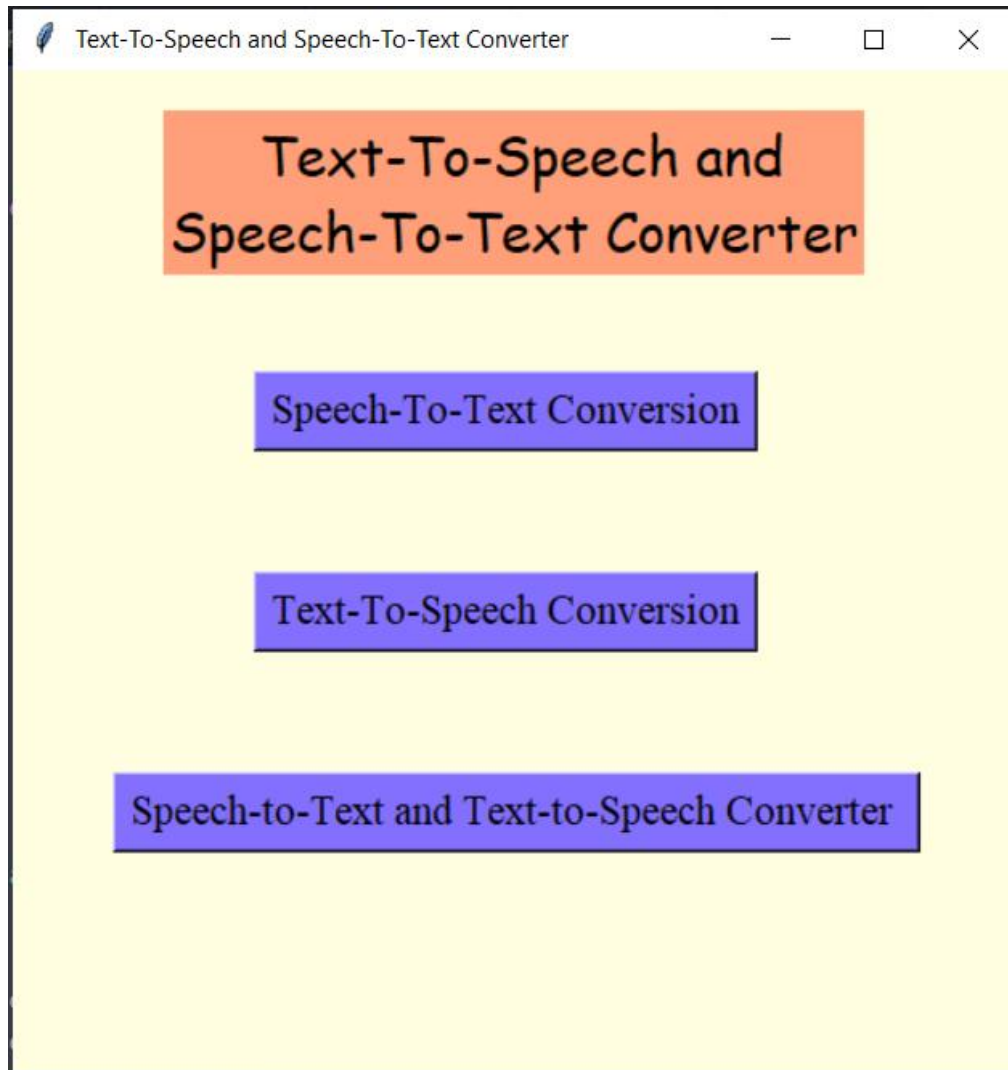
Results and Discussions:

- **Different pages of the Graphical User Interface;**
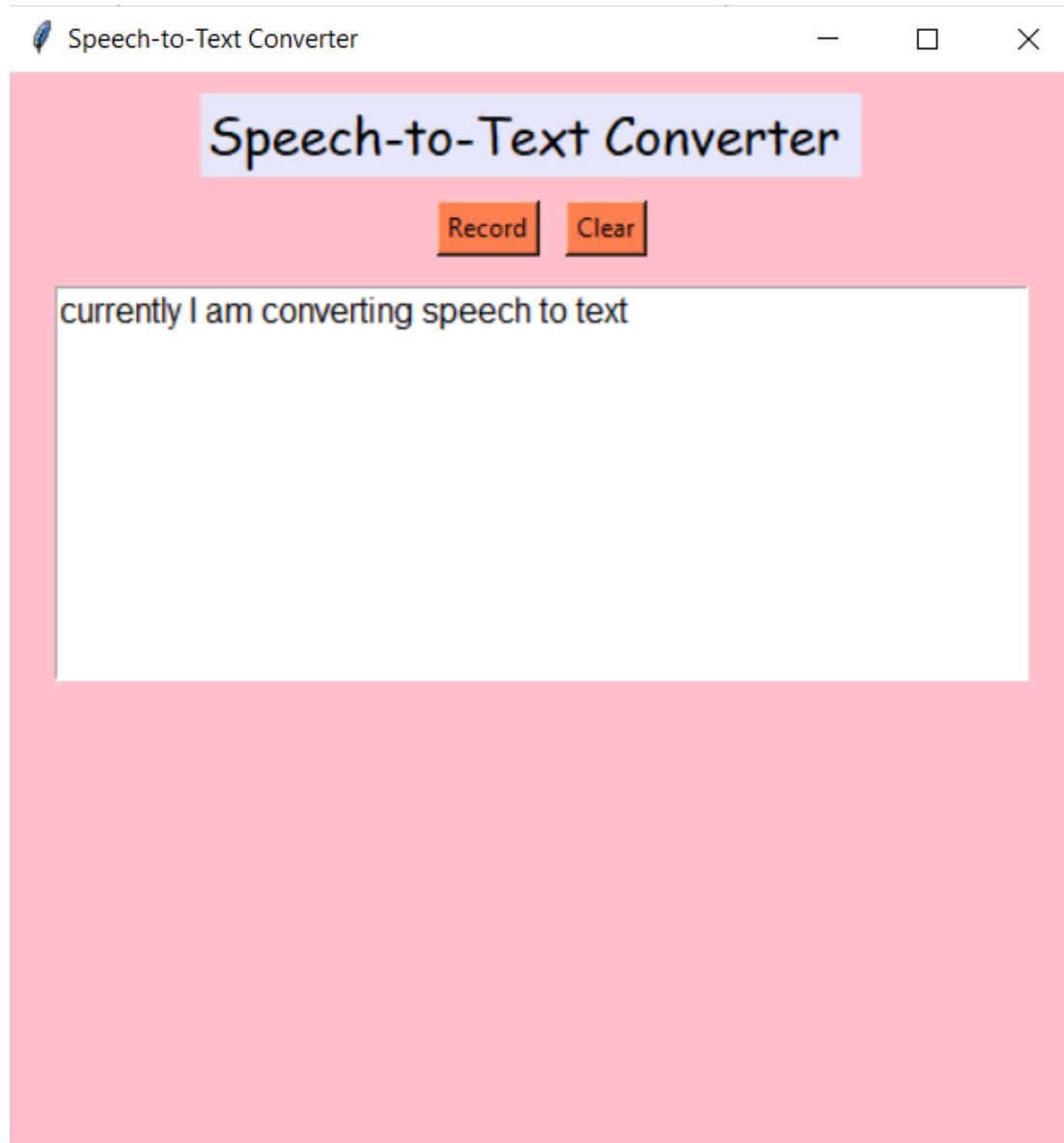
  - ➢ **Main – Interface:**

    It contains the access to all the three pages of the interface namely – Speech-To-Text, Text-to-Speech and combination Speech-To-Text and Text-to-Speech designed to complete the three different tasks respectively.

Figure. shows the main page – that contains links to the three different interfaces.

➢ Speech-to-Text Converter Interface:

It contains Record button to record the voice of the user and a Clear button to clear the output text from the window. For speech to text conversion the user clicks on the record button. The speech is then converted into text and displayed on the text box.
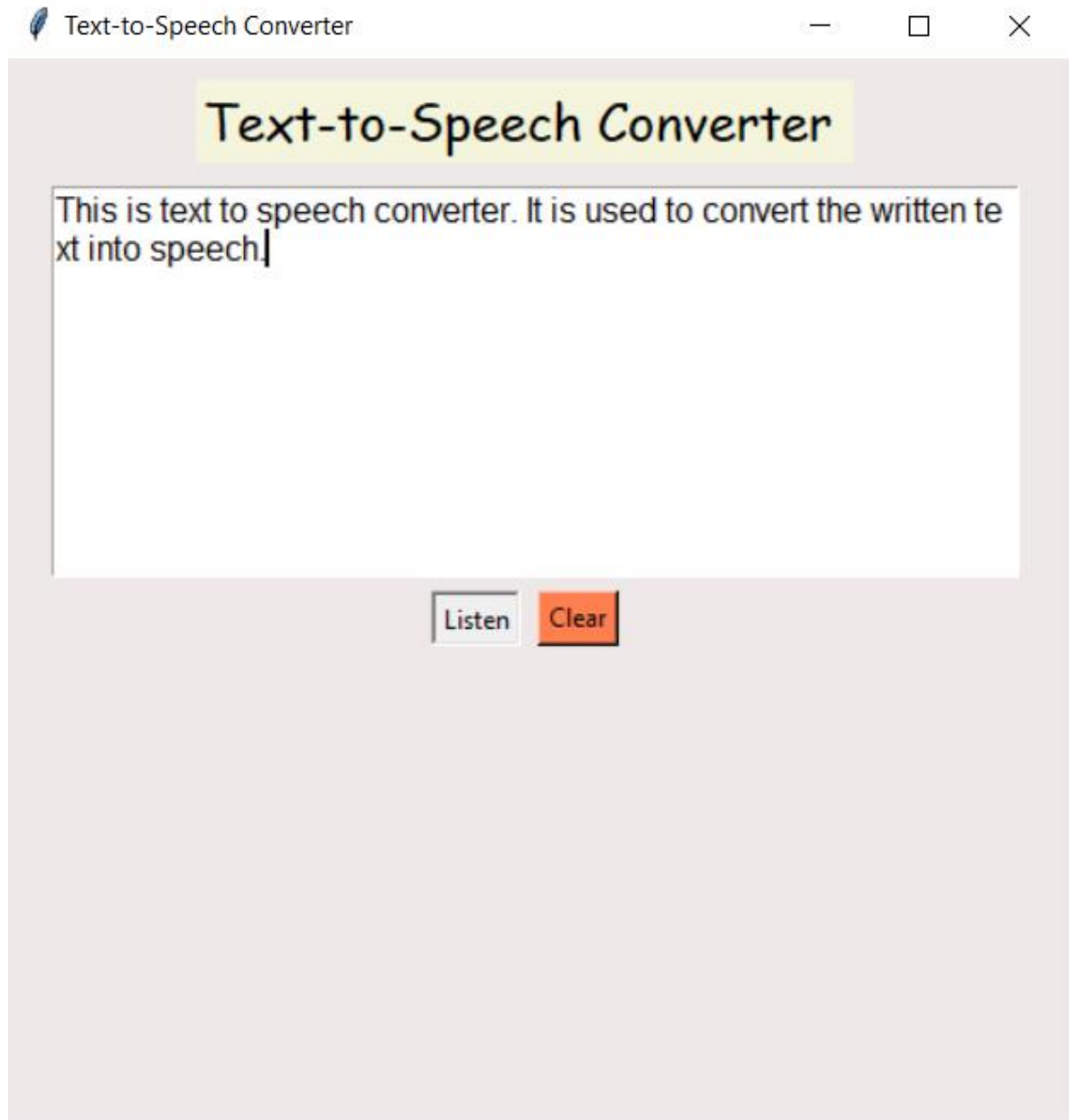

Figure. shows the Speech – To – Text Interface which converts the Speech input into text.

➢ **Text-To-Speech Converter Interface:**

It contains a Listen button to listen the output and a Clear button to clear the output text.

To convert text into speech the user types the text content in the text window and then clicks on the Listen button to listen to the output.
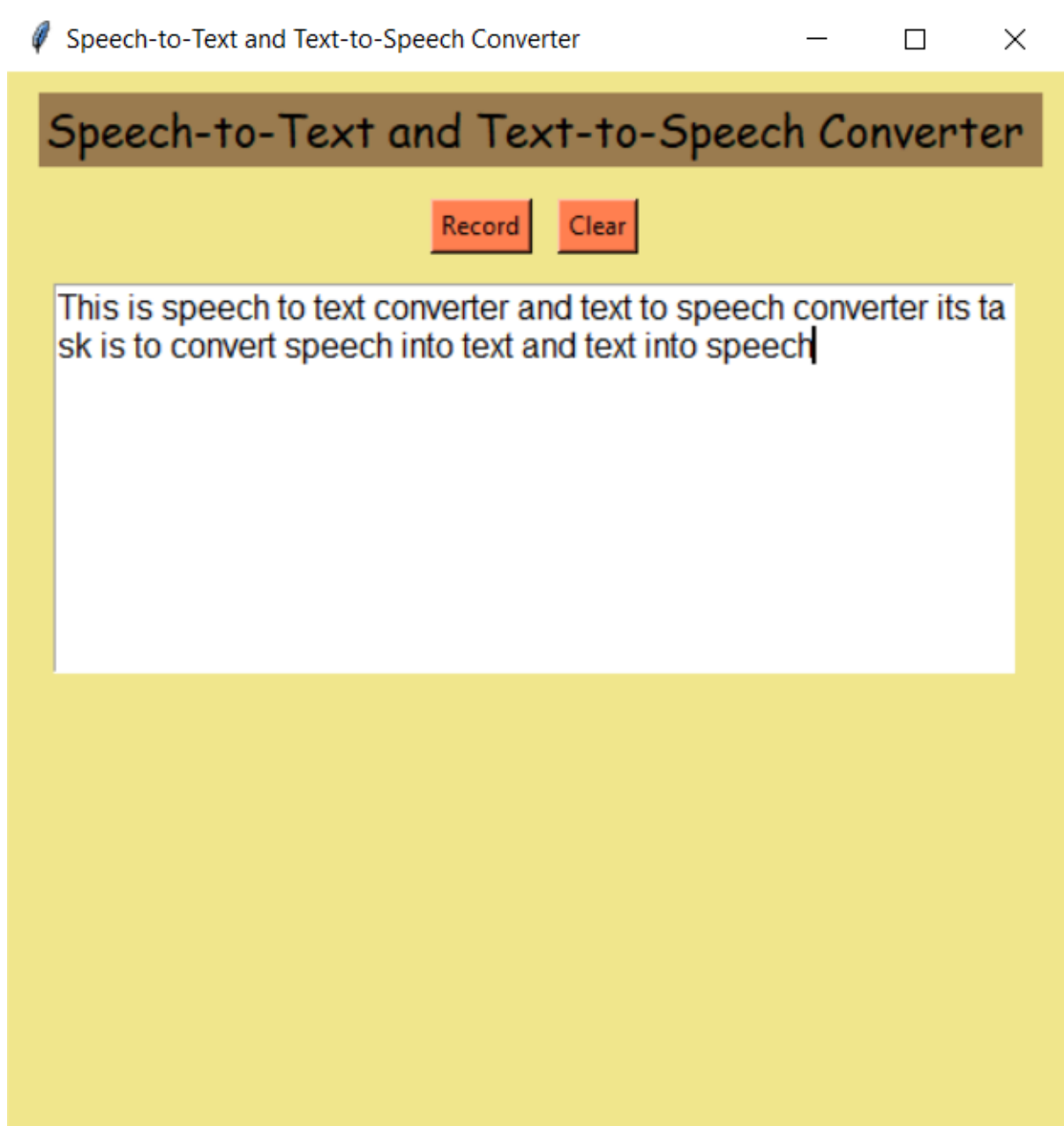

Figure. shows. the Text – To – Speech Interface which converts the typed text into speech.

➢ **Speech-To-Text and Text-To-Speech Converter Interface:**

It contains a Record button to start recording the speech and Clear button to clear the output window.

As soon as the user clicks on the Record button the speech by the user is then converted into text and displayed in the text box. The same text is also narrated by the system.

Figure. shows the Speech-To-Text and Text-To-Speech Converter Interface.

References:

1. https://docs.python.org/3/library/tkinter.html
2. https://betterprogramming.pub/an-introduction-to-pyttsx3-a-text-to-speech-converter-for-python-4a7e1ce825c3
3. "Python GUI - tkinter - GeeksforGeeks" https://www.geeksforgeeks.org/python-gui-tkinter/amp/
4. https://stackabuse.com/introduction-to-speech-recognition-with-python/
5. https://www.geeksforgeeks.org/python-convert-speech-to-text-and-text-to-speech/?ref=rp
6. https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/#:~:text=pyttsx3%20is%20a%20text%2Dto,a%20reference%20to%20a%20pyttsx3.
7. https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/#:~:text=pyttsx3%20is%20a%20text%2Dto,a%20reference%20to%20a%20pyttsx3.
8. "Speech Recognition" https://pypi.org/project/SpeechRecognition/