AVN INSTITUTE OF ENGINEERING & TECHNOLOGY
Accredited by NAAC & NBA

MINI PROJECT

BY BATCH -3

NALLA VAISHNAVI : 195U1A0577

VADLA RAKESH KUMAR : 195U1A05B0

NALLAGASU SHIVAKUMAR : 195U1A0578

MANIDEEPAK : 195U1A0567

***UNDER THE GUIDANCE OF: Dr.* P. MEENA KUMARI** (PROFESSOR)

TITLE of the project :

SPEECH TRANSLATOR & text translator

ABSTRACT:

- The developing technology has a lot of influence on us making us choose **simpler and easier ways to do tasks**. One of them is using the speech option **rather than typing**
- Here we are **with a Python** to convert the speech to text and vice versa.
- This is a project that provides the user with **multiple options**, one to convert text to speech and the other to convert speech to text and various other features.
- For the **first case**, the user enters the text and then gets to listen to it.
- And in the **latter case**, the user speaks and then gets the text shown on the window.
- Python provides many **Modules / libraries** to convert text to speech.

PROBLEM STATEMENT:

- The problems occurred before having translater includes:

➢ More **man power** is required to prepare documentation.

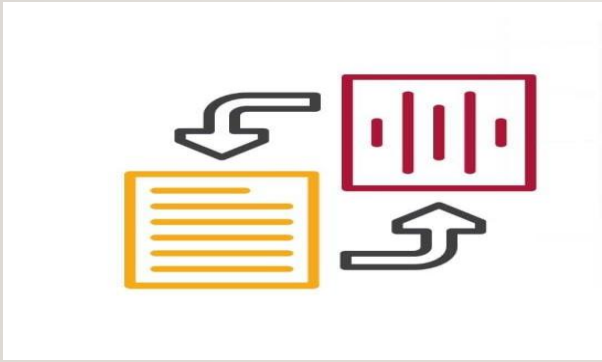
➢ More **time** is consumed

➢ **Disabled people** are **not able** to communicate with the world at ease.

»If we can implement Translator Mechanism then , we can easily over come the problems mentioned above

AIM:

- To **translate** the text entered by user into speech.
- To **recognise** the words spoken by a user and display them accordingly as a text



OBJECTIVE:

- The main objective of our project is to create modern technology appreciation and awareness by computer operators, to develop our own converter of speech to text and text to speech (this is a 2 way mission).
- This helps in **reducing the time** taken for documentation in modern world, with less man power too.
- To enable the **deaf and dumb** to communicate and contribute to the growth of an organization through synthesized voice , to enable the **blind and elderly** people enjoy a User-friendly computer interface.

SCOPE:

- The study is focused on an ideal combination of a human-like behaviour with computer application to build a one-way interactive **medium** between **the computer and the user.**
- Speech recognition is an evolving technology. It is one of the many ways people can **communicate with computers with little or no typing**
- Today, in this age of globalization, the scope of translation is immense. It stretches from our immediate environment to every sphere of life. The significance and relevance of translation in our daily life is multidimensional and extensive.

EXISTING SYSTEM:

- We have many **alternatives** and **techniques** to translate the text to **speech** and speech recognition operation.
- Out of all these available systems , the accuracy of the output is most important thing , which is very less in most of the applications , this has a result people don't show much interest in this type of applications .

PROPOSED SYSTEM:

- The proposed system is designed to **eliminate the disadvantages** of the existing system
- The proposed system "**SPEECH TRANSLATOR & text translator**" will

  increase the accuracy rate
- The system can **defeat the constant challenges** of unskilled individuals and improve their way of life.
- Proposed system will provide a **user friendly** environment to the users to perform their tasks such as text translation to **speech** and listen it easily.

ADVANTAGES:

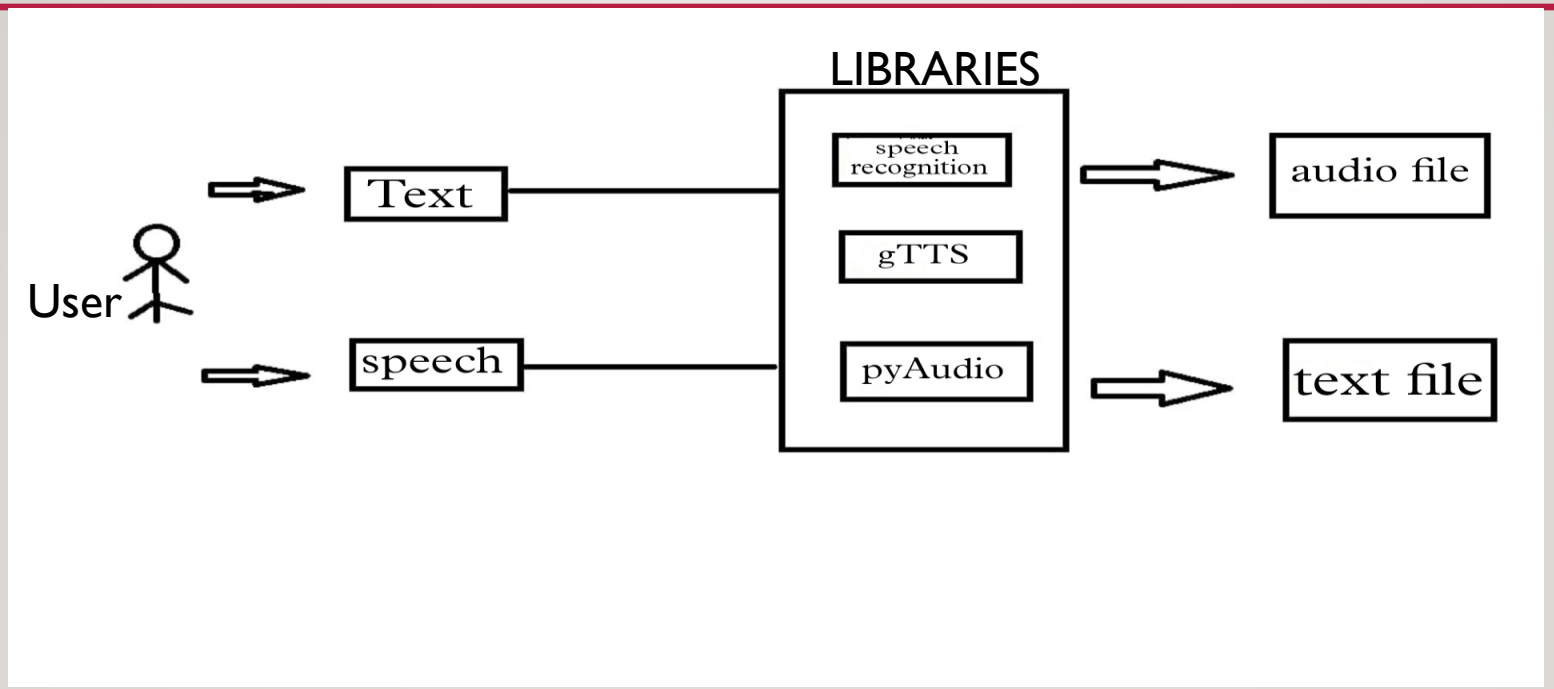
- **Work on the go**
  - ✓ Speech-to-text translator software enables you and your employees to work on the go, further increasing productivity and efficiency
- **Improve your organization's accessibility**
  - ✓ Incorporating speech-to-text translation technology into your business operations will make your organization a more accessible one (includes colleges , and all other documentation places)
- **Help people with disabilities**
- **Improved accuracy**
  - ✓ The best speech-to-text translation software can now provide you with higher accuracy rates.

REQUIRMENTS:

- **SOFTWARE REQUIRMENTS**:
  - Python(programming language)
  - Adding dependency(Libraries){ pyttsx3 , pyAudio, speech_recognition }

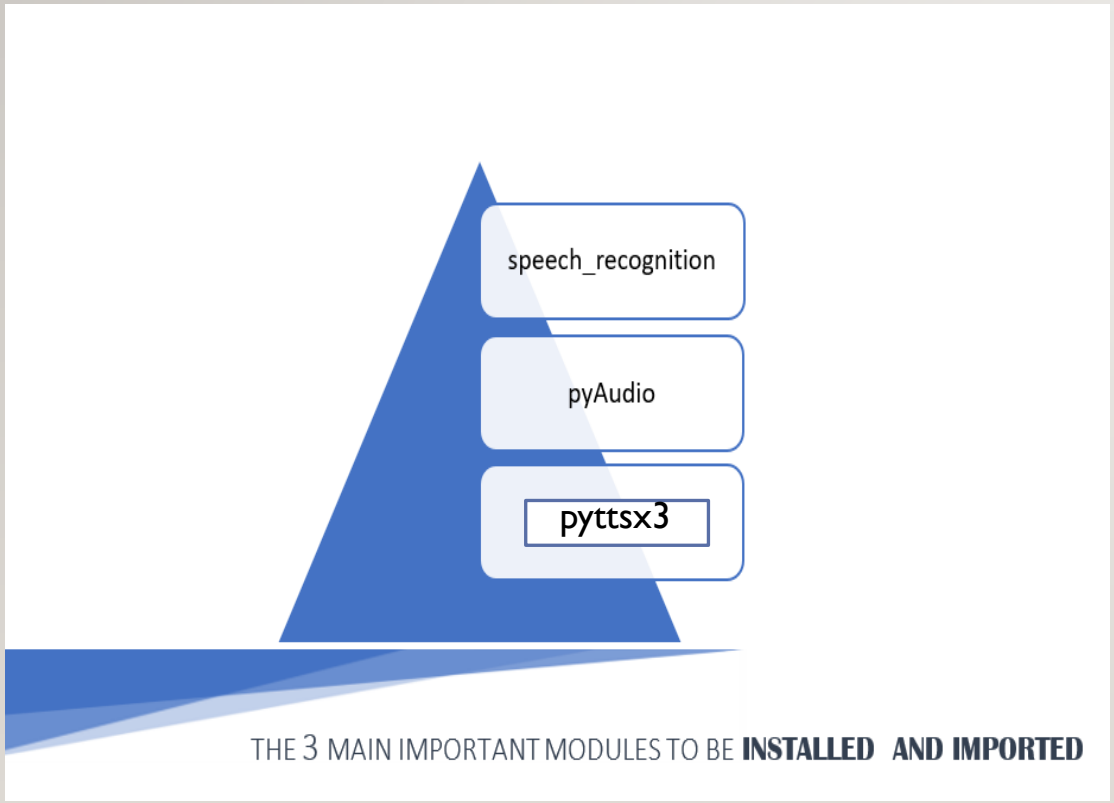
- **HARDWARE REQUIRMENTS:**
  - System : minimum of i3 Processor:
  - Input Devices : Keyboard , Microphone , Mouse
  - RAM : minimum 4 GB
  - Storage : (based on user requirements)

ARCHITECTURAL DESIGN:



- **User** can give **input** in the form of **text** or a **voice**, which is then **processed** using the appropriate **libraries** to its respective output i.e., **audio file** or **a text file.**

LOGICAL DESIGN:



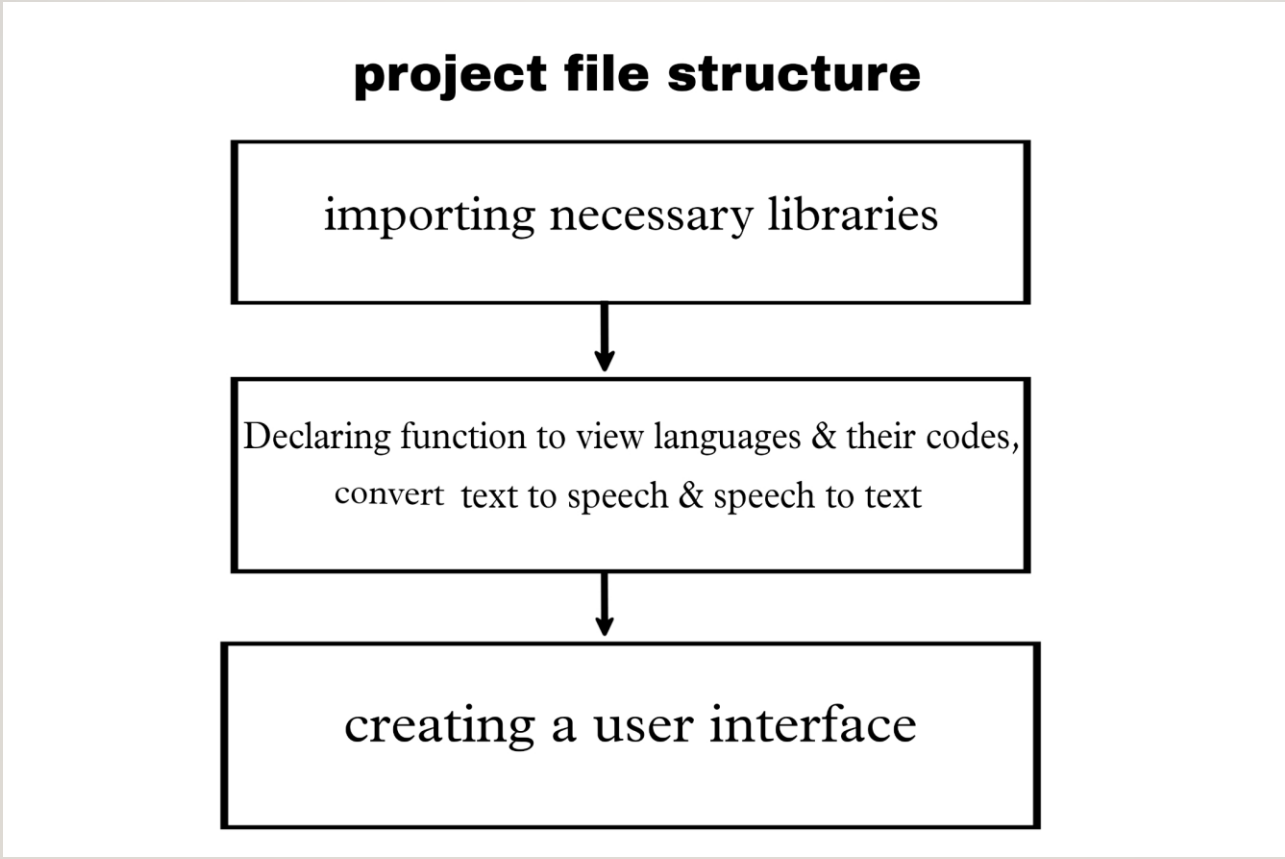
pyttsx3
library
convert text to speech

speech recognition
module
convert speech to text

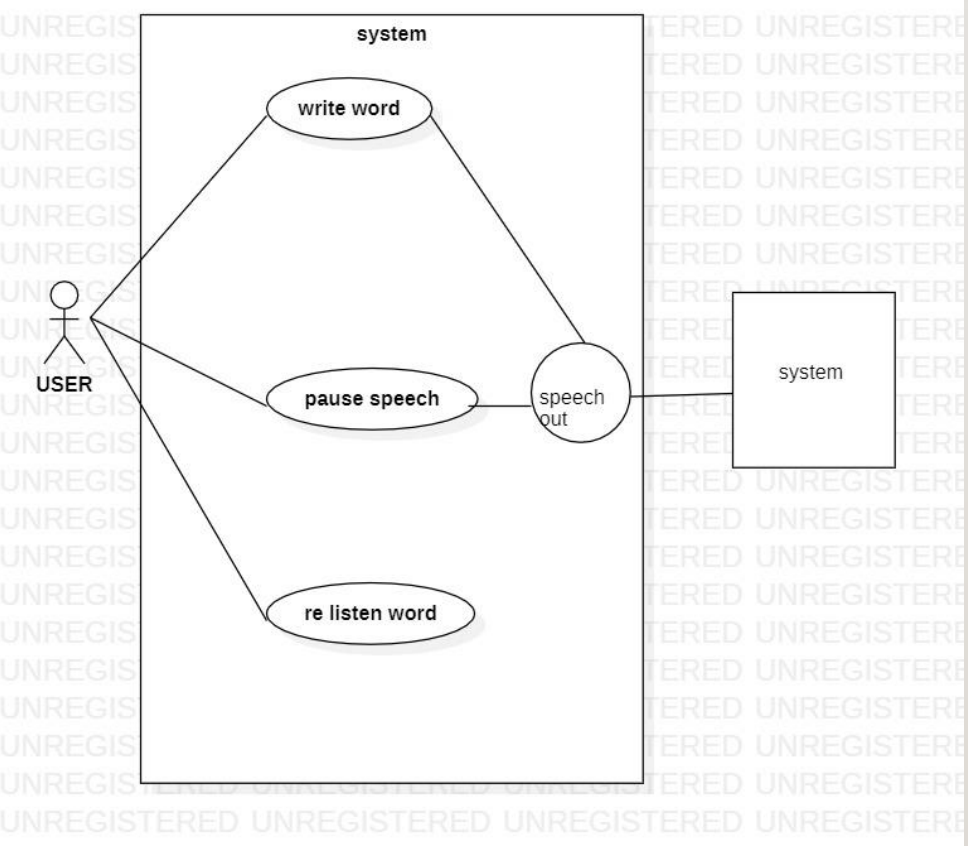
tkinter
contain widgets & support to receive user input
creates user interface

PyAudio
module
to play and record audio

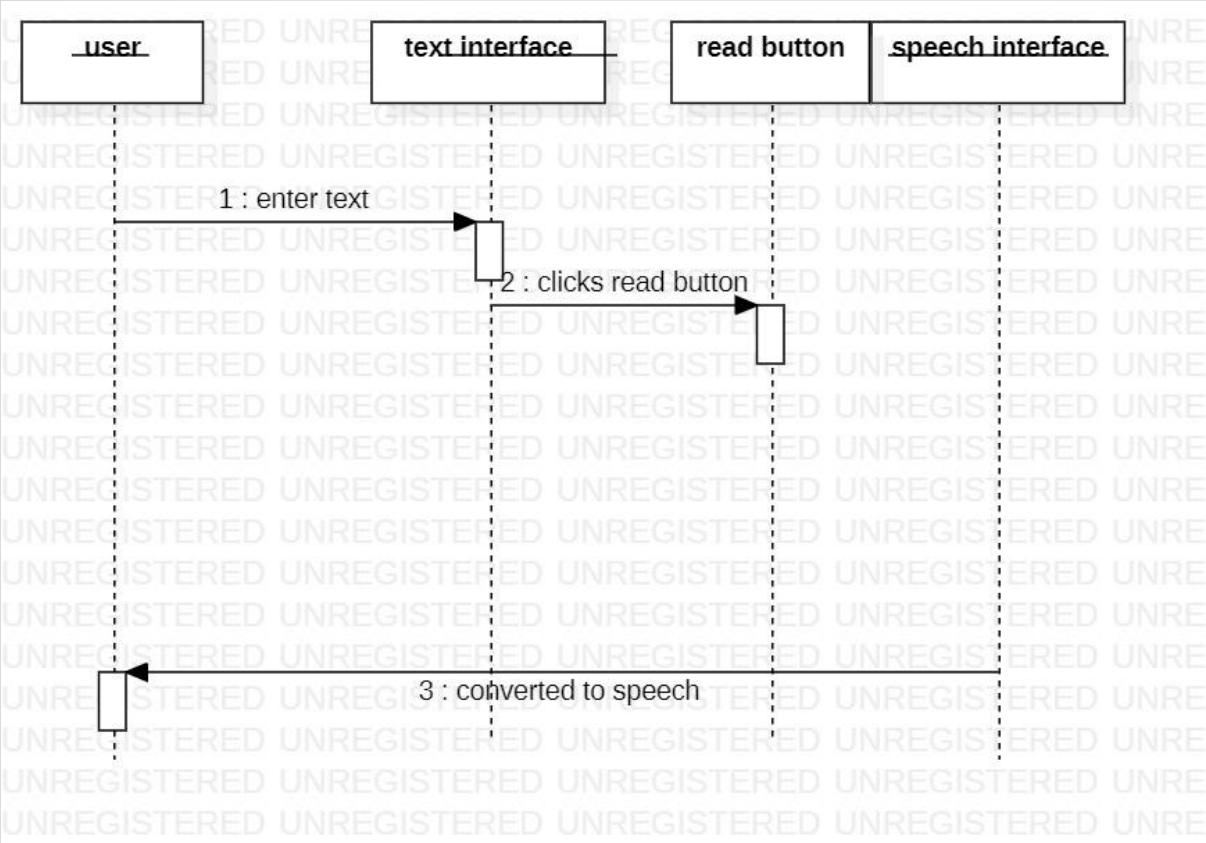
DESIGN CONCEPTS:



USE CASE DIAGRAM:

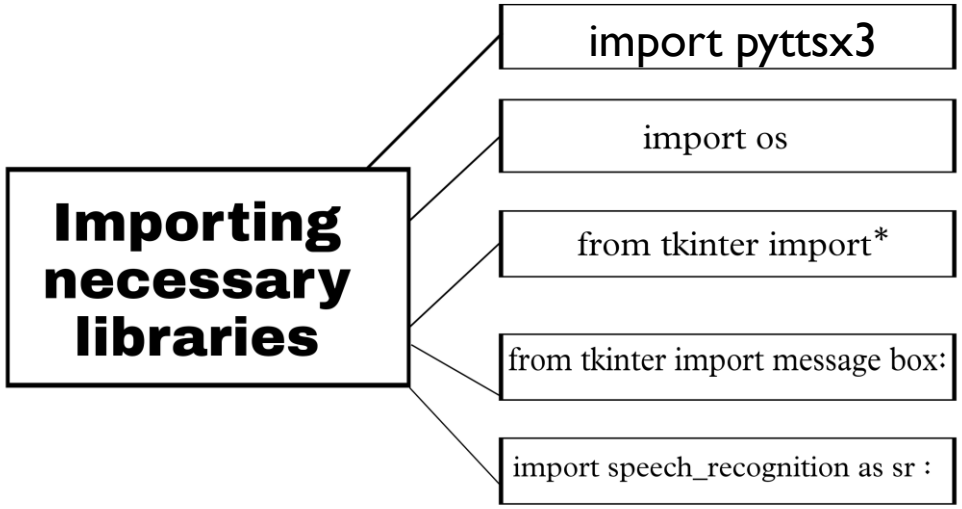


SEQUENCE DIAGRAM:



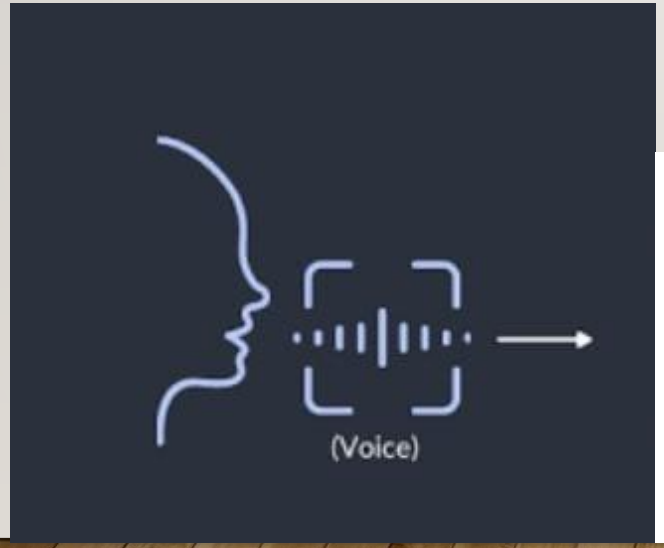
CONCEPTUAL DESIGN:

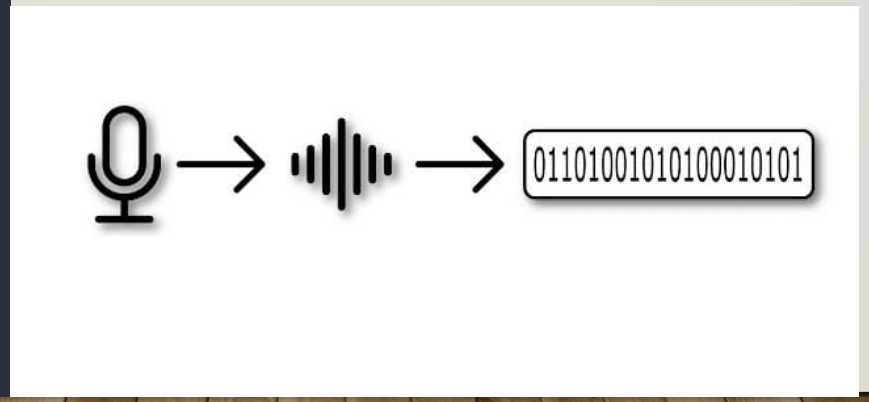
- This part mainly **consists of importing** the required libraries after the installation of the main modules
- tkinter here refers to – to create a **user interface** later, we initially import it too.



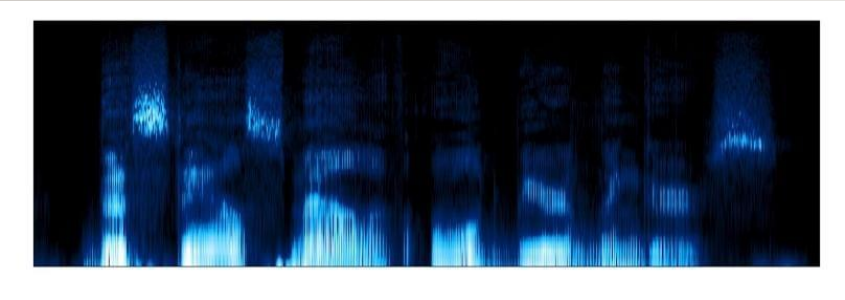
HOW EACH MODULE WORKS INTERNALLY:

- Speech must be **converted** from a **physical signal** to an **electrical signal** using a **microphone**
- And then to **digital data** using an **Analogue -to-Digital converter**
- Once digitized several models can be used to transcribe the audio to text(**HMM**)

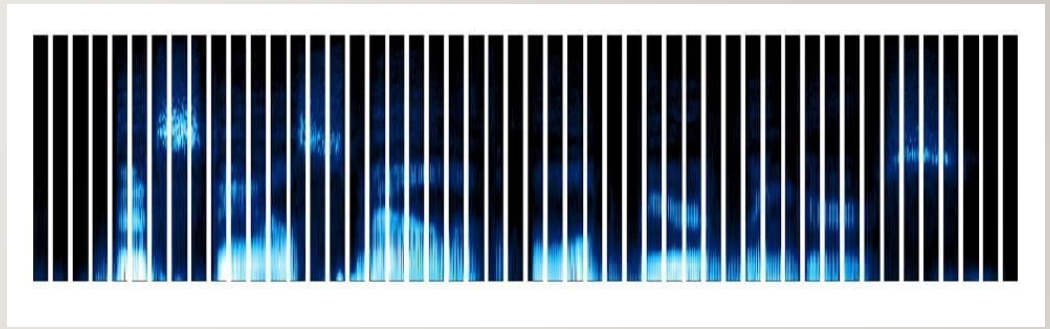




HMM WORKING:

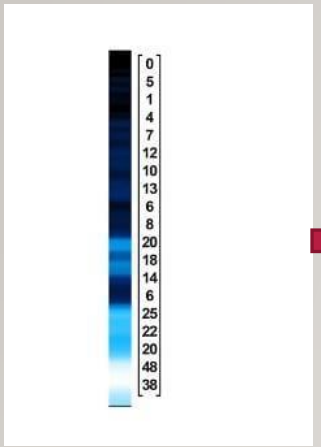


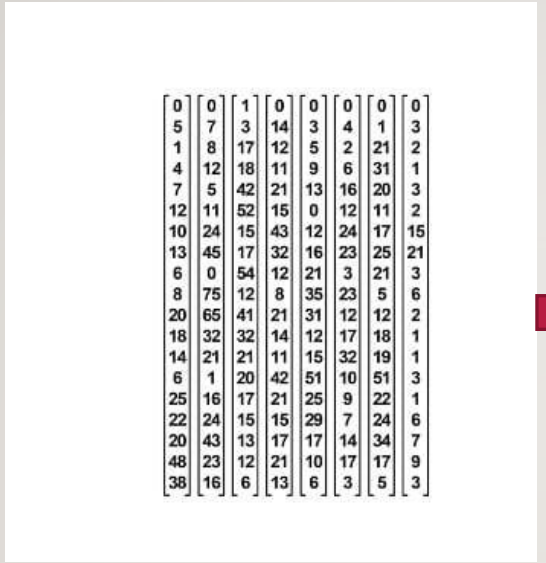
Original speech signal

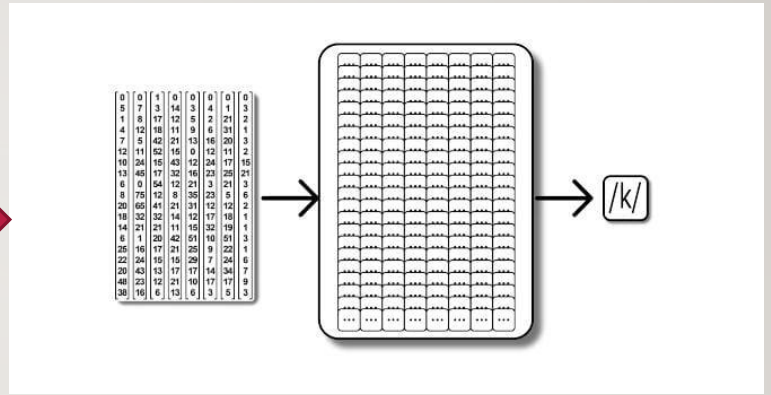


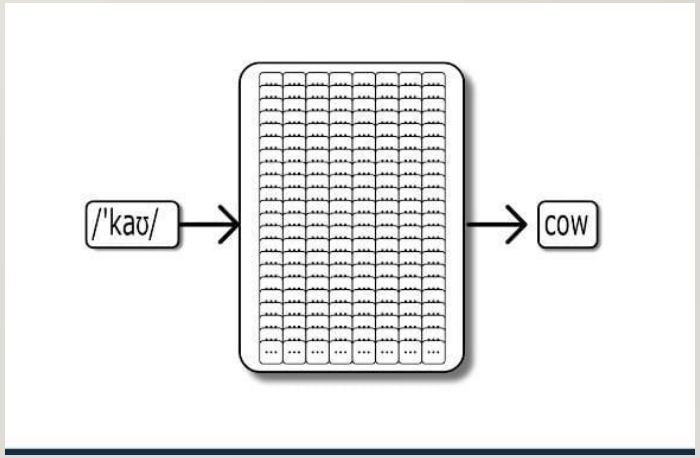
Split into segments

- The speech signal is divided into 10-millisecond fragments at first and later next division starts at 8th part of the previous divisions which prevents the data leakage.
- **Power spectrum** of each fragment is calculated, is mapped to vectors
- The final **output** of HMM is a **sequence of the vectors**
- To convert speech to text, a group of vectors is **matched** to one or more phonemes – a fundamental unit of speech.


0
5
1
4
7
12
10
13
6
8
20
18
14
6
25
22
20
48
38


0 0 1 0 0 0 0 0
5 7 3 14 3 4 1 3
1 8 17 12 5 2 21 2
4 12 18 11 9 6 31 1
7 5 42 21 13 16 20 3
12 11 52 15 0 12 11 2
10 24 15 43 12 24 17 15
13 45 17 32 16 23 25 21
6 0 54 12 21 3 21 3
8 75 12 8 35 23 5 6
20 65 41 21 31 12 12 2
18 32 32 14 12 17 18 1
14 21 21 11 15 32 19 1
6 1 20 42 51 10 51 3
25 16 17 21 25 9 22 1
22 24 15 15 29 7 24 6
20 43 13 17 17 14 34 7
48 23 12 21 10 17 17 9
38 16 6 13 6 3 5 3


/k/


/ˈkaʊ/
cow

TEXT-TO-SPEECH:

- **Tokenize**: Tokenize a sentence into words
- **Phonemes/Pronunciation**: It breaks input text into phonemes, based on their pronunciation.

For example,

- "**Hello, Have a good day**" converts to **HH AH0 L OW1, HH AE1 V AH0 G UH1 D D EY1.**

- **Phoneme duration:** Represents the total time taken by each phoneme in the audio.

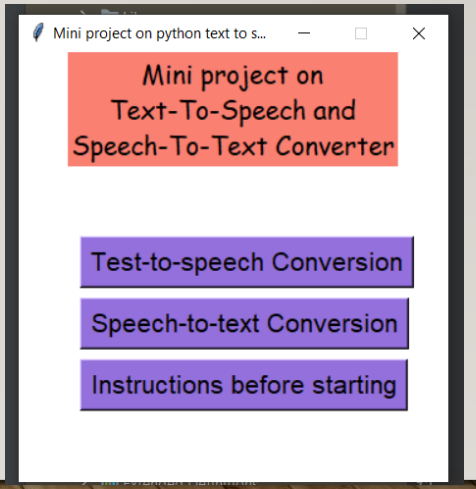
CODING PHASE:

SAMPLE CODE:

```python
# Importing all the necessary modules
from tkinter import *
from tkinter.messagebox import showinfo
import pyttsx3
import speech_recognition as sr
```

```python
# Creating the main GUI window
root = Tk()
root.title('Mini project on python text to speech and speech to text Converter')
root.geometry('350x350')
root.resizable(0, 0)
root.configure(bg='White')

# Placing all the components
Label(root, text='Mini project on Text-To-Speech and Speech-To-Text Converter',
      font=('Comic Sans MS', 16), bg='Salmon', wrap=True, wraplength=300).place(x=40, y=0)

tts_btn = Button(root, text='Test-to-speech Conversion', font=('Helvetica', 16), bg='MediumPurple', command=TTS)
tts_btn.place(x=50, y=150)

stt_btn = Button(root, text='Speech-to-text Conversion', font=('Helvetica', 16), bg='MediumPurple', command=STT)
stt_btn.place(x=50, y=200)

instruction_btn = Button(root, text='Instructions before starting', font=('Helvetica', 16), bg='MediumPurple',
                         command=instruction)
instruction_btn.place(x=50, y=250)
```

Reference for explanation:



TESTING PHASE:

UNIT TESTING:

- ✓In the unit testing we **test each module** individually and integrate with the overall system.
- ✓Unit testing focuses verification efforts on the smallest unit of software design in the module.
- ✓ The **module** of the system is **tested separately**.
- ✓ This **testing** is carried out during programming stage itself.
- ✓There are some **validation** checks for fields also.
- ✓ **For example** the validation check is done for **varying** the **user input** given by the user which validity of the data entered. It is very easy to find **error** debut the system.

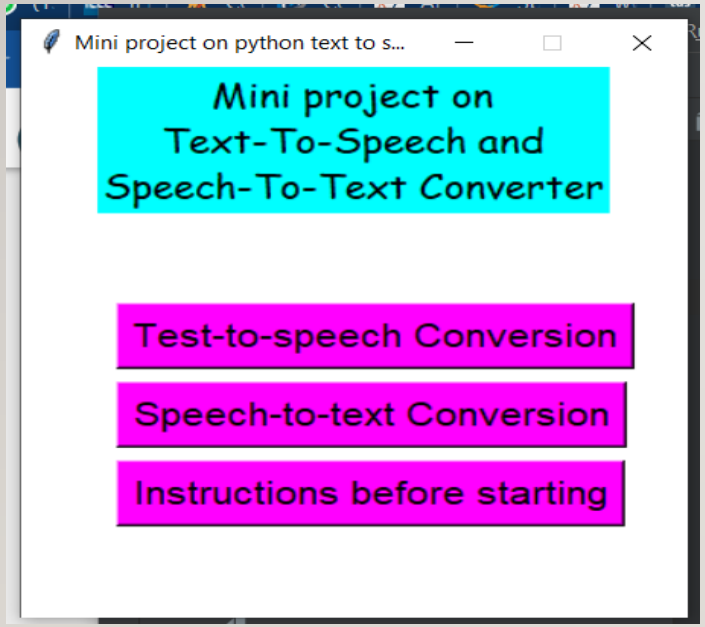
TEST CASES:

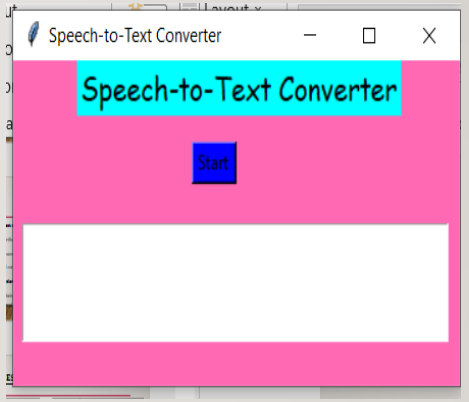
# Test Case 1 :	TTS
Name of Test:	Text to speech
Items being tested:	Message box, Play audio button , outed audio.
Sample Input:	Taken from random user: ex: hi this is our mini project
Expected output:	All words of users should be played as an audio
Actual output:	Total data is played.
Remarks:	**Pass.**

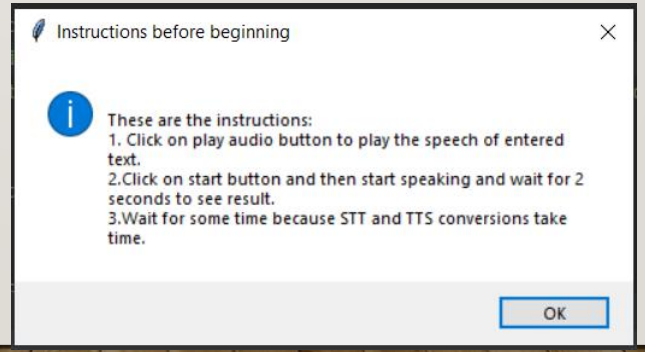
# Test Case 2 :	STT
Name of Test:	Speech to text
Items being tested:	Start button, microphone access, outed text.
Sample Input:	Taken from random user: ex: Hello how are you good morning
Expected output:	All words that a user speaks should be displayed in dialog box.
Actual output:	Total data is displayed.
Remarks:	**Pass.**

RESULTS:

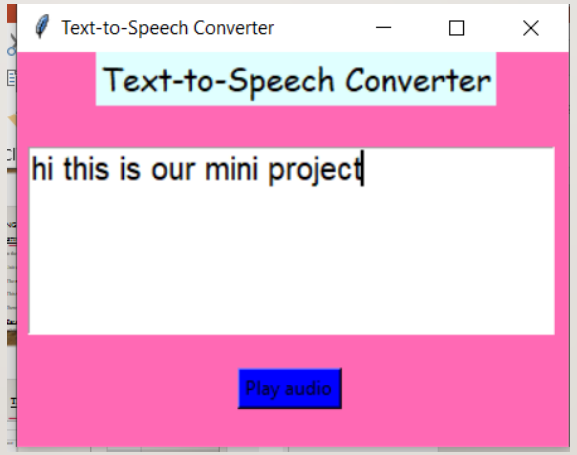
SAMPLE OUTPUT:



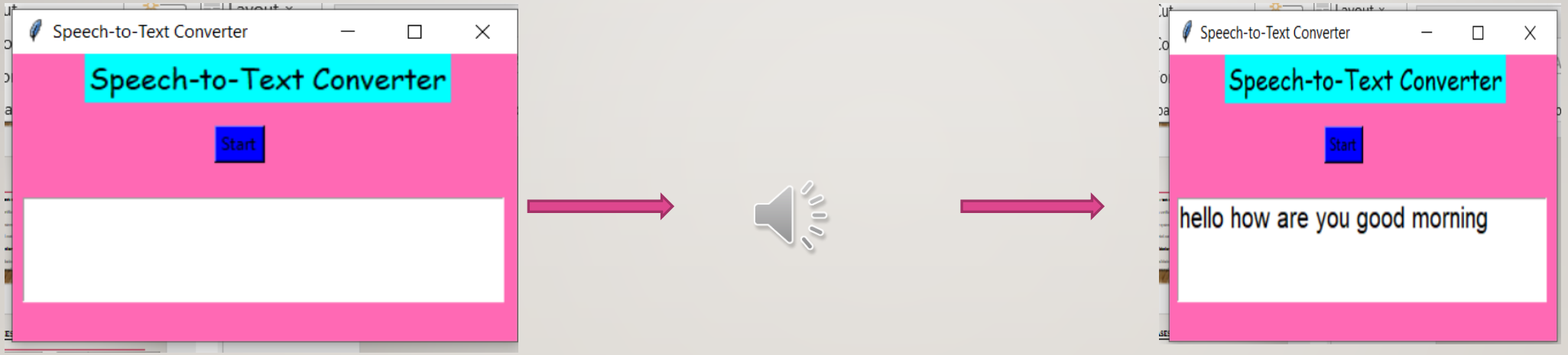




RESULTS: (SAMPLE OUTPUT)



RESULTS: (SAMPLE OUTPUT)



CONCLUSION:

- Finally it may be concluded that the developed system is with good accuracy and at the same time it is **more efficient** , as we are **developing a place** where the **user** can easily interact with the system interface and **fulfil their requirements**.
- We are now able to convert text to speech and speech to text using a tkinter window using a simple python code.
- There is a scope for further future developments in our project which will help the people in a immense way.

THANK YOU