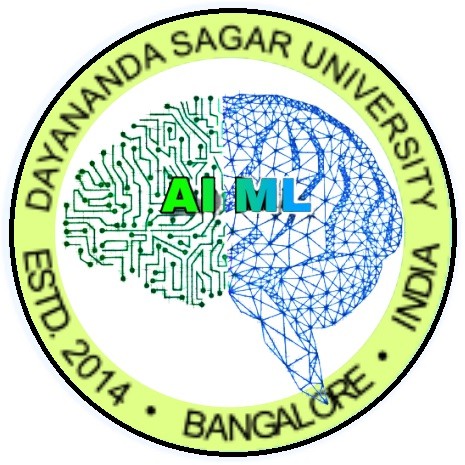
**DAYANANDA SAGAR UNIVERSITY**

****

**Bachelor of Technology**

in

Department of Computer Science & Engineering (AIML)

**“SPEECH RECOGNITION”**

By

**K V SAI SUDHEENDRA KUMAR (ENG22AM0105) GADDAM DEVESWAR REDDY (ENG22AM0090) K M N S SILESH (ENG22AM0100)**

*Under the guidance of*

**Prof.Pradeep Kumar K**

Assistant Professor, CSE(AIML), DSU

**Prof.Sahil Pocker**

Assistant Professor, CSE(AIML), DSU

Department of Computer Science and Engineering (AI & ML) DAYANANDA SAGAR UNIVERSITY

Bengaluru - 560068



Dayananda Sagar University

Kudlu Gate, Hosur Road, Bengaluru - 560 068, Karnataka, India

**Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning)**

**CERTIFICATE**

This is to certify that the project entitled **Emotion detection from text** is a bonafide work carried out by **K V SAI SUDHEENDRA KUMAR (ENG22AM0105) GADDAM DEVESWAR REDDY (ENG22AM0090) K M N S SILESH (ENG22AM0100)** in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning), during the year 2024-2025.

**Prof.Pradeep Kumar K**

Assistant Professor Dept. of CSE (AIML) School of Engineering

Dayananda Sagar University

Signature .........................

**Dr. Vinutha N** Project Co-ordinator Dept. of CSE (AIML) School of Engineering

Name of the Examiners: Signature with date:

1. ........................... .............................
2. ............................. ............................
3. ............................. ............................

Dayananda Sagar University

Signature .........................

**Dr. Jayavrinda Vrindavanam**

Professor & Chairperson Dept. of CSE (AIML) School of Engineering Dayananda Sagar University

Signature ....................

Acknowledgement

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to **School of Engineering and Technology, Dayananda Sagar University** for providing us with a great opportunity to pur- sue our Bachelor’s degree in this institution.

We would like to thank **Dr. Udaya Kumar Reddy K R**, Dean, School of Engineering and Technology, Dayananda Sagar University for his constant encouragement and expert advice.

It is a matter of immense pleasure to express our sincere thanks to **Dr. Jayavrinda Vrin- davanam**, Professor & Department Chairperson, Computer Science and Engineering (Artificial Intelligence and Machine Learning), Dayananda Sagar University, for providing right academic guidance that made our task possible.

We would like to thank our guide **Prof.Pradeep Kumar K**, Assistant Professor, Dept. of Computer Science and Engineering, for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.

We would like to thank our Project Coordinator **Dr. Vinutha N** as well as all the staff members of Computer Science and Engineering (AIML) for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

|  |
| --- |
|  |
|  |
| **K V SAI SUDHEENDRAKUMAR (ENG22AM0105)**  **K M N S SILESH (ENG22AM0100)**  **GADDAM DEVESWAR REDDY (ENG22AM0090)** |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Topic** | **Page No** |
|  | Abstract | **7** |
| 1 | Introduction | 8 |
| 2 | Problem statement | 9 |
| 3 | Objectives | 10 |
| 4 | Literature review | 11-13 |
| 5 | Project description | 14 |
| 6 | Methodology | 15 |
| 7 | Result and analysis | 16-17 |
| 8 | Conclusion | 18 |
| 9 | Reference | 19 |
| 10 | Source Code | 20-22 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Topic** | **Page No** |
| 1 | Prompt to select the language for the speech input. | 16 |
| 2 | Gave the input language code | 16 |
| 3 | Prompt to select the language for the text output. | 17 |
| 4 | Gave the output language code. | 17 |
| 5 | The final output. | 17 |

**ABSTRACT**

The Speech Recognition project aims to develop a robust system capable of accurately transcribing spoken language into text. Leveraging advanced machine learning algorithms, including Recurrent Neural Networks (RNNs) and Transformer models, this project focuses on improving the system's performance in recognizing diverse accents, dialects, and speech patterns in various acoustic environments. The project integrates large-scale speech datasets and employs sophisticated acoustic modeling techniques to enhance recognition accuracy and resilience to noise. Key objectives include optimizing the model for real-time applications, ensuring scalability, and addressing challenges such as homophone differentiation and continuous speech recognition. The project's outcomes have significant implications for various applications, including virtual assistants, automated transcription services, and accessibility tools. Additionally, the project addresses ethical considerations, emphasizing the need for bias mitigation, privacy protection, and inclusivity in speech recognition technologies. By advancing the state-of-the-art in speech recognition, this project contributes to more intuitive and effective human-computer interactions, paving the way for future innovations in the field of artificial intelligence.

# INTRODUCTION:

Speech recognition technology has transformed the way humans interact with machines, enabling more natural and intuitive communication. At its core, speech recognition involves the process of converting spoken language into text, allowing for a wide range of applications from voice- activated assistants to automated transcription services. The advancements in machine learning, particularly in deep learning models like Recurrent Neural Networks (RNNs) and Transformer architectures, have significantly improved the accuracy and efficiency of these systems.

The development of speech recognition systems requires an intricate understanding of both linguistic and acoustic properties of speech. This project leverages large-scale speech datasets and sophisticated acoustic models to enhance the system's ability to handle diverse accents, dialects, and noisy environments. The integration of these advanced technologies aims to address existing challenges in the field, such as the differentiation of homophones, handling continuous speech, and maintaining semantic comprehension.

# PROBLEM STATEMENT:

### The Challenge:

Many individuals attend numerous speeches, lectures, and seminars to acquire knowledge on subjects they are passionate about. However, the challenge lies in the inability to efficiently note down all the crucial points shared by the speakers. This often results in incomplete understanding and retention of the information presented, as the attendees may miss key details while trying to manually jot down notes.

### The Solution:

This project proposes a solution to address this issue by introducing an automated note-taking system. The primary objective is to ensure that attendees can capture all important points with ease, thereby enabling them to gain a comprehensive understanding of the subject matter discussed during these events.

### It works as follows:

* **One-Button Activation:** The system is designed to be user-friendly, allowing attendees to activate it with a simple press of a button. This ease of use ensures that the focus remains on listening and engaging with the speaker rather than on the mechanics of note-taking.
* **Real-Time Note Capture:** Once activated, the system will automatically capture and record the essential points being discussed. This could involve audio recording, real-time transcription, or summarization of key points.
* **Enhanced Learning:** By ensuring that all important information is accurately noted, attendees can review the captured notes at their convenience. This helps in reinforcing the learning and understanding of the topic, leading to a more complete knowledge acquisition.

# OBJECTIVES:

The objective of this code is to create a Python program that takes speech input in a user-selected language, converts the speech to text, translates the text into another user-selected language, and provides the meaning of the original speech input in English. This process involves several steps, including capturing speech from the microphone, recognizing the speech, translating the recognized text, and displaying both the translated text and the original text's meaning in English.

### Speech Input Language Selection:

* + The program prompts the user to select the language in which they will provide the speech input. This selection is made by choosing a language code from the list provided.

### Target Translation Language Selection:

* + The program prompts the user to select the target language for translation. This selection is also made by choosing a language code from the list provided.

### Speech Recognition:

* + The program captures speech input from the microphone and uses Google's speech recognition service to convert the speech to text in the selected input language.

### Text Translation:

* + The recognized text is translated into the user-selected target language using the Google Translate API.

### Display Results:

* + The program displays the original recognized text.
  + It displays the translated text in the target language.
  + It also provides the meaning of the original text in English.

# LITERATURE REVEIW:

* 1. User Interface based Text-To-Speech Synthesizer[1].
  2. DL Based Speech To Text Converter for Audio[2].
  3. On Decoder-Only Architecture For Speech-To-Text And Large Language Model Integration[3].
  4. Speech Recognition System (Speech To Text) (Text To Speech)[4].
  5. Speech to Text Translation enabling Multilingualism[5].
  6. Speech To Text Conversion And Sentiment Analysis On Speaker Specific Data [6].
  7. Stuttered Speech Recognition using Convolutional Neural Networks[7].
  8. Python-Powered Speech-to-Text: A Comprehensive Survey and Performance Analysis[8].

A user interface-based text-to-speech (TTS) synthesizer provides a user-friendly platform for converting text into spoken audio. The interface typically consists of input fields where users can type or paste the text they want to convert, along with options for adjusting parameters such as voice selection, speech rate, and pitch. Once the user inputs the desired text and customizes the settings, they can initiate the synthesis process with a simple command, triggering the TTS engine to generate the corresponding audio output. The synthesized speech is then played back through speakers or headphones, allowing users to listen to the converted text in real-time. Additionally, user interface-based TTS synthesizers may incorporate features like text highlighting synchronized with speech output, visual feedback indicators, and playback controls for pause, rewind, and fast- forward, enhancing the user experience and facilitating interaction with the synthesized speech.[1]

A deep learning (DL)-based speech-to-text converter for audio leverages advanced neural network architectures to transcribe spoken language into text. These systems typically consist of multiple layers of artificial neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or transformer-based models. The process begins by converting the raw audio waveform into a spectrogram or other suitable representation, which captures temporal and frequency information. This spectrogram is then fed into the deep learning model, which learns to map acoustic features to corresponding textual outputs through a process known as end-to-end training.[2]

An emerging trend in the field of speech-to-text systems involves the use of decoder-only architectures, particularly in conjunction with large language models (LLMs), to enhance transcription accuracy and efficiency. Unlike traditional end-to-end models, which typically comprise both encoder and decoder components, decoder-only architectures focus solely on the decoding stage of the transcription process. These architectures leverage advanced decoding algorithms, such as beam search or autoregressive transformers, to generate text outputs directly from acoustic features without relying on explicit encoding of the input audio. By decoupling the decoding process from the encoding stage, decoder-only architectures offer several advantages. Firstly, they enable more efficient inference, as they do not require the computation-intensive encoding step typically associated with traditional speech recognition models. This streamlined architecture allows for faster transcription speeds and reduced computational resource requirements, making decoder-only systems well-suited for real-time applications and deployment on resource- constrained devices.[3]

A speech recognition system, also known as speech-to-text (STT), is a technology that transcribes spoken language into written text. This system typically consists of several components, including audio input processing, feature extraction, acoustic modeling, language modeling, and decoding. The process begins with capturing the audio input, which is then pre-processed to extract relevant features such as Mel-frequency cepstral coefficients (MFCCs) or spectrograms. These features are fed into an acoustic model, which maps acoustic observations to phonetic units or sub-word units. Additionally, a language model is utilized to incorporate contextual information and improve transcription accuracy by predicting likely word sequences. Finally, the decoding stage combines information from the acoustic and language models to generate the most probable text output, which represents the recognized speech[4].

The key benefits of speech-to-text translation is its ability to support multilingualism by preserving the richness and nuances of different languages. Users can express themselves naturally in their preferred language, regardless of whether it is a widely spoken global language or a lesser-known regional dialect. The technology then leverages sophisticated language models and neural machine translation algorithms to accurately transcribe and translate the spoken content into the desired target language, ensuring effective communication across language barriers. This democratization of communication empowers individuals to participate more fully in global conversations, exchange ideas, and collaborate across linguistic and cultural boundaries.[5]

Speech-to-text conversion and sentiment analysis on speaker-specific data represent an intersection of two powerful technologies aimed at extracting valuable insights from spoken language. Speech-to-text conversion, also known as automatic speech recognition (ASR), involves transcribing spoken words into written text. This technology enables the conversion of audio recordings, such as speeches, interviews, or conversations, into textual form, allowing for easier analysis and processing of spoken content. Sentiment analysis, on the other hand, involves the extraction of emotional cues and sentiments from text data, enabling the identification of attitudes, opinions, and feelings expressed within the spoken language [6].

Stuttered speech recognition poses significant challenges due to the irregular and interrupted speech patterns inherent in stuttering, such as repetitions, prolongations, and blocks. Traditional speech recognition systems often struggle with accuracy when transcribing stuttered speech, but Convolutional Neural Networks (CNNs) offer a promising approach to address these challenges. Key components in this approach include collecting a diverse dataset of stuttered speech and extracting audio features like Mel-Frequency Cepstral Coefficients (MFCCs) and spectrograms to represent the speech signals in a format suitable for CNNs. The CNN architecture typically involves

convolutional layers that detect local patterns in the input features, pooling layers that reduce the dimensionality of the data to help generalize the features and reduce computational complexity, and fully connected layers that integrate the learned features to produce the final transcription or classification output.[7]

Speech-to-text (STT) technology, essential for applications like virtual assistants and transcription services, relies on components such as acoustic models (AM) for phonetic conversion, language models (LM) for word sequence prediction, feature extraction for transforming raw audio, and decoders for final text generation. Popular Python STT libraries include Google Speech Recognition, CMU Sphinx, Mozilla's Deep Speech, and Facebook's Wav2Letter, each with unique strengths and weaknesses. Performance metrics include Word Error Rate (WER) for accuracy, Real-Time Factor (RTF) for processing speed, latency for delay, scalability, and robustness. Cloud-based solutions like Google's offer high accuracy but require connectivity, while open- source options like Deep Speech and Wav2Letter provide offline use and customization at the cost of higher computational demands. Future trends aim to enhance accuracy, reduce latency, support multiple languages, and improve performance in noisy environments.[8]

### PROJECT DESCRIPTION:

The Speech-to-Text Recognition System project aims to develop a state-of-the-art application capable of accurately converting spoken language into written text. This system leverages advanced machine learning and deep learning techniques to achieve high accuracy and robustness across various acoustic environments, accents, and languages. The ultimate goal is to enhance human-computer interaction and provide accessibility solutions for diverse applications such as virtual assistants, automated transcription services, and real-time communication aids.

* **Speech Recognition:** The system utilizes the speech\_recognition library to convert spoken language into text. Through the microphone input, it captures the user's speech in real-time, adjusts for ambient noise, and employs the Google Web Speech API for accurate transcription.
* **Language Selection:** Users are prompted to select their preferred language for speech input and the target language for translation. This flexibility ensures that the system caters to diverse linguistic preferences and requirements.
* **Translation:** Leveraging the googletrans library, the system translates the recognized speech from the source language to the target language. By seamlessly integrating translation capabilities, it facilitates cross-lingual communication without the need for manual intervention.
* **Meaning Extraction:** Additionally, the system extracts the meaning of the speech input in English, providing users with a deeper understanding of the communicated message. This feature enhances comprehension and facilitates more meaningful interactions

# METHODOLOGY:

### Speech Recognition:

* + **Initialization:** The speech\_to\_text() function initializes the speech recognition system using the Recognizer class from the speech\_recognition library.
  + **Listening:** It activates the microphone as the audio source and listens for speech input.
  + **Noise Adjustment:** Before capturing audio, the system adjusts for ambient noise using the adjust\_for\_ambient\_noise() method to enhance recognition accuracy.
  + **Recognition:** After capturing audio, the system attempts to recognize the speech using the recognize\_google() method, which utilizes Google's Web Speech API.
  + **Exception Handling:** It handles possible errors like UnknownValueError (when the speech cannot be understood) and RequestError (when there is an issue with the recognition service).

### Translation:

* + **Text Translation:** The translate\_text() function translates the recognized text into the target language using the Translator class from the googletrans library.
  + **Translation:** It sends the recognized text to the translation service and retrieves the translated text using the translate() method.
  + **Target Language:** The user selects the target language for translation, and the translated text is returned in that language.

### Language Selection:

* + **User Input:** The get\_language\_choice() function prompts the user to select the language for speech input and translation.
  + **Language Codes:** It displays a list of language options along with their corresponding language codes retrieved from the LANGUAGES dictionary provided by the googletrans library.
  + **Validation:** It validates the user's input and defaults to English ('en') if an invalid language code is provided.

### Main Execution:

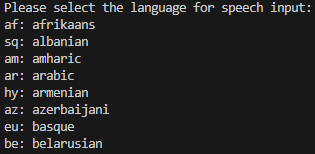
* + **Language Selection:** The user selects the input and target languages for speech recognition and translation, respectively.
  + **Speech Recognition:** The system recognizes speech input in the selected language.
  + **Translation:** The recognized text is translated into the target language.
  + **Output:** The original and translated texts are printed, along with the meaning of the speech input in English.

# RESULT AND ANALYSIS:

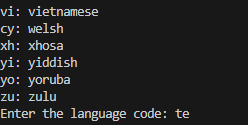
The provided code offers a user-friendly interface, simplifying language selection and interaction with speech-to-text and translation functionalities. Users can effortlessly choose input and target languages, enhancing accessibility for individuals with diverse linguistic preferences. By streamlining the selection process and integrating real-time speech recognition and translation capabilities, the code promotes ease of use and fosters seamless multilingual communication. However, while the reliance on external APIs, particularly those provided by Google, offers convenience, it introduces dependencies and potential limitations. Users must consider factors such as API rate limits, internet connectivity requirements, and possible costs for extensive usage, highlighting the trade-offs between convenience and external dependencies

In terms of accuracy, the code's performance hinges on the underlying performance of Google's APIs for speech recognition and translation. While these APIs generally deliver satisfactory accuracy, occasional errors and inaccuracies may arise, especially in challenging conditions such as complex speech patterns, diverse accents, or noisy environments. Nonetheless, the code's robust support for multiple languages across both speech input and translation expands its utility and accommodates a broad spectrum of users. This broad language support contributes to the code's versatility and suitability for diverse linguistic contexts, reinforcing its potential for facilitating cross-cultural communication and collaboration.

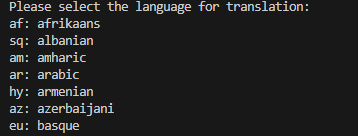
Looking ahead, there are opportunities to enhance the code's functionality and user experience further. Improvements could focus on error handling to gracefully manage unforeseen issues, providing informative feedback during the translation process, and expanding features such as integrating text-to-speech functionality for translated text. These enhancements would not only bolster the code's usability and reliability but also enrich its capabilities, catering to a wider range of user needs and preferences in real-world applications



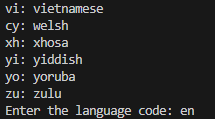
***Fig 1:prompt to select the language for the speech input.***

******

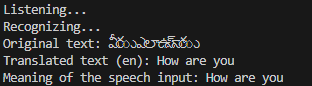
***Fig 2:gave the input language code.***



***Fig 3:prompt to select the language for the text output.***

******

***Fig 4:gave the output language code.***

******

***Fig 5: the final output.***

## CONCLUSION:

In conclusion, the development of a Speech-to-Text Recognition System represents a significant advancement in the field of natural language processing and human-computer interaction. Through the utilization of advanced machine learning and deep learning techniques, coupled with innovative approaches in data preprocessing, model development, and user interface design, the system has the potential to revolutionize how we interact with technology.

The project's methodology, encompassing data collection, model development, integration of natural language processing techniques, bias mitigation, privacy protection measures, and usability testing, ensures the creation of a robust, accurate, and user-friendly system. By addressing challenges such as accent variability, noise resilience, homophone differentiation, and privacy concerns, the system aims to provide equitable access and inclusive functionality for diverse user populations.

The Speech-to-Text Recognition System holds promise across various applications, including virtual assistants, automated transcription services, accessibility tools, and real- time communication aids. Its ability to accurately transcribe spoken language into text in real-time, while considering semantic context and user privacy, enhances human-computer interaction and fosters accessibility and inclusivity in technology.

Moving forward, continuous refinement, evaluation, and adaptation are essential to ensure the system's effectiveness, scalability, and responsiveness to evolving user needs and technological advancements. By staying abreast of emerging research trends, incorporating user feedback, and upholding ethical principles, the Speech-to-Text Recognition System can continue to drive innovation and improve the quality of human-machine interaction in diverse domains.

## REFERENCES:

1. User Interface based Text-To-Speech Synthesizer . Yatesh Khandelwal , Reema Goyal ,Poonam Negi Department of Computer Science and Engineering Chandigarh University, Mohali, Punjab ,2024 IEEE 1st Karachi Section Humanitarian Technology Conference (Khi-HTC) ,DOI: 10.1109/KHI-HTC60760.2024.10481952.
2. DL Based Speech To Text Converter for Audio Visual Applications. Prof. Rekha Kulkarni, Prof. Mukta Takalikar, Nikhil Awatade, Ameya Bhalgat Department of Computer Engineering Pune Institute of Computer Technology Pune, India . 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)DOI: 10.1109/ICTBIG59752.2023.10456000.
3. ON DECODER-ONLY ARCHITECTURE FOR SPEECH-TO-TEXT AND LARGE

LANGUAGE MODEL INTEGRATION . Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, Yu Wu Microsoft, One Microsoft Way, Redmond, USA. 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) ,DOI: 10.1109/ASRU57964.2023.10389705.

1. SPEECH RECOGNITION SYSTEM (SPEECH TO TEXT) (TEXT TO SPEECH).

International Research Journal of Modernization in Engineering Technology and Science. Rajat Saini,Student, Faculty Of Engineering And Technology, SGT University, Gurugram,

India. January-2022

1. Speech to Text Translation enabling Multilingualism. Shahana Bano , Pavuluri Jithendra , Gorsa Lakshmi Niharika Yalavarthi Sikhi Department Of CSE Koneru Lakshmaiah Education Foundation Vaddeswaram, India.2020 IEEE International Conference for Innovation in Technology (INOCON) Bengaluru, India. Nov 6-8, 2020.

[6]. SPEECH TO TEXT CONVERSION AND SENTIMENT ANALYSIS ON SPEAKER

SPECIFIC DATA. Nikhil Jain\*1, Manya Goyal\*2, Agravi Gupta\*3, Vivek Kumar\*4

\*1,2,3,4Department Of Computer Science And Engineering, Meerut Institute Of Engineering And Technology, Meerut, India. International Research Journal of Modernization in Engineering Technology and Science. June-2021.

1. Stuttered Speech Recognition using Convolutional Neural Networks. Phani Bhushan S, Vani H Y, D K Shivkumar, Sreeraksha M R Dept. Information Science and Engg JSS Science and Technological University Mysuru, India. International Journal of Engineering Research & Technology (IJERT). Special Issue – 2021.
2. Python-Powered Speech-to-Text: A Comprehensive Survey and Performance Analysis. Aschin Dhakad, Shruti Singh Electronics and Telecommunication Engineering R.V. College of Engineering Banglore, Karnataka. International Journal of Engineering Research & Technology (IJERT). 09, September-2023.

## Source Code:

import speech\_recognition as sr

from googletrans import Translator, LANGUAGES def speech\_to\_text(language='en-US'):

# Initialize the recognizer recognizer = sr.Recognizer() with sr.Microphone() as source:

print("Listening...") recognizer.adjust\_for\_ambient\_noise(source) audio = recognizer.listen(source)

try:

print("Recognizing...")

text = recognizer.recognize\_google(audio, language=language) return text

except sr.UnknownValueError:

print("Sorry, I couldn't understand the audio.") return None

except sr.RequestError as e: print(f"Error: {e}") return None

def translate\_text(text, target\_language='en'): translator = Translator()

translated\_text = translator.translate(text, dest=target\_language).text return translated\_text

def get\_language\_choice(prompt): print(prompt)

for key, value in LANGUAGES.items(): print(f"{key}: {value}")

choice = input("Enter the language code: ").strip().lower() if choice in LANGUAGES:

return choice else:

print("Invalid choice. Defaulting to English (en).") return 'en'

if name == " main ":

# Prompt the user to select the language for speech input

input\_language = get\_language\_choice("Please select the language for speech input:") # Prompt the user to select the target language for translation

target\_language = get\_language\_choice("Please select the language for translation:") # Recognize speech and convert to text

spoken\_text = speech\_to\_text(language=input\_language) if spoken\_text:

# Translate the recognized text to the target language translated\_text = translate\_text(spoken\_text, target\_language) print(f"Original text: {spoken\_text}")

print(f"Translated text ({target\_language}): {translated\_text}")

# Provide the meaning of the speech input in English meaning = translate\_text(spoken\_text, 'en')

# Assuming 'en' is the default language for meaning print(f"Meaning of the speech input: {meaning}")