SpeakEZ: AI Language Translation Application

Kaali Lovell, Pavan Poliready, Rakesh Tirumala,
Steve St Fleur, Tanvi Prakash Gavali, Venkat Sai Pallapu
Seidenberg School of Computer Science and Information Systems
Pace University, New York, NY, USA

Abstract— SpeakEZ is an artificial intelligent web application designed to create seamless communication between individuals of different speaking languages. Due to the current climate of wars between countries and the open border policies, there have been a migration of people with different linguistic backgrounds, seeking to find a better life from their current volatile locations. SpeakEZ hopes to address this issue by allowing users to input audio and allowing them to transcribe or translate the audio into their preferred language. SpeakEZ uses advanced AI models to accurately transcribe the audio and produce real-time transcriptions. With this, SpeakEZ hopes to break the communication barriers between those of different languages and cultures.

***Keywords*— artificial intelligence, machine learning, encoder-decoder, neural networks, transcription, translation, seq2seq, SMT, NMT, NLP, WER**

I. INTRODUCTION

With the recent global conflicts around the world in areas such as Isreal, Palestine, Ukraine, Mexico, and Haiti, this has led to a mass migration of individuals relocating to seek a better life. There are currently over 7,000 living languages worldwide and with this migration, individuals of different language backgrounds are constantly encountering one another. Being able to communicate with each other is important for sharing ideas, information, and fostering global understanding amongst one other.

To close the communication gap and provide ease of communication between individuals, this study introduces SpeakEZ, an AI- integrated application for language transcription and translation. SpeakEZ allows users to upload audio files, or record audio using our platform. SpeakEZ also allows the transcription of audio in its original language input and the translation of the transcription into the user's preferred language. The application will also allow users to store or download their transcribed audio and translations securely for future use, if they are account owners with SpeakEZ.

The security and ease of access with using SpeakEZ, makes it a reliable tool for reducing language barriers between individuals and promoting inclusivity between users.

II. LITERATURE REVIEW

A. *Current Solutions*

"Google Translate is a free online service that allows global users to translate text into a desired language."[6] Google Translate has been in service since 2006 and has hence implemented different models over time to produce translations for users. As described in [3], initially, Google Translate used statistical machine translations (SMT) to assist with the generations of translations. Later on, however, Google Translate implemented neural machine translation (NMT) to help produce translations.

SMT uses Bayes Theorem. Bayes Theorem is a mathematical formula for determining conditional probabilities. "Conditional probability assesses the likelihood of an event occurring based on prior events or data." [2] In the context of Bayes Theorem with respect to translation, the theorem suggests that a sentence in one language corresponds to a sentence in another language. The SMT model breaks the input text into smaller segments, and matches each segment with its equivalent translation in its targeted language. It then reconstructs the sentence back together. In order to decide the correct translated equivalent, the SMT approach uses conditional probability as mentioned before.[3] We will explain this approach using the following example. If within the database containing a large dataset of words, if there has been a large subset of the word "hola" being matched to the word "hello," then the probability of the word "hola" being matched to the word "hello" would be higher than any other Spanish translated word version. This example proves the idea of conditional probability- the likelihood of the event "hola" being chosen based on prior events occurring, "the constant matching of hola being matched to hello in a large dataset".

Advancements in AI have allowed us to introduce neural network models into our applications, such as Google Translate who implemented the model later on. NMT produces more accurate translations and takes up less space, hence less costs. NMT differs from SMT in that NMT does not break down the sentences, rather it tries to translate the sentences as a whole using "deep learning neural networks" to translate the work. NMT models are trained on large datasets of human-translated text and create contextual representations of words or phrases, allowing for the model to return results in a human-like way. Neural networks are trained to think like a human brain to allow the results produced to seem more human-like. The dataset of human-translated text would then in turn produce a translation result that would be more accurate to the way a human would respond in the targeted language.

Although Google Translate has implemented more advanced models over time, their results are still limited and produce less accurate outcomes. Google Translate tends to mistranslate colloquial terms, and rare spoken languages. "Research shows that native speakers often find its output comprehensible but not flawless, highlighting the need for further editing to achieve accuracy.” [12]

B. Relevancy

SpeakEZ differs from Google Translate in that it focuses on recognizing audio rather than using a text-based approach. Due to this, SpeakEZ accepts audio input rather than Google Translate which uses text inputs. SpeakEZ’s model also focuses on encoding and decoding which ensures accuracy in transcription and translation within the application. Users can upload audio files or record audio using the platform. They are also able to transcribe the audio in the spoken language, as well as translate the transcription into their chosen language. The model implemented within SpeakEZ updates it database daily with over 680,000 hours of supervised multilingual data sourced from the web. This increases accuracy and reliability of the results produced through SpeakEZ. The constant improvements through the database will allow users to communicate with confidence to others of a different language background.

This study is valuable for users of different backgrounds- it will facilitate ease of communication between individuals of different languages. By using audio as inputs, language translation is more efficient and faster compared to traditional typed inputs for translation.

III. PRODUCT REQUIREMENTS

Our application serves as an effective platform for users to seamlessly communicate with one another. To develop the SpeakEZ application, we integrate the following key features:

1) Functional Requirements

The following are functional requirements for the SpeakEZ application:

1. User Authentication: The application should allow users to use their credentials to login or create an account with SpeakEZ. SpeakEZ should have a login screen to allow users to atleast enter their email address and password to login.
2. Audio Input: The application should be able to accept audio of an .mp3 format, whether it be imported from the users’ device, or recorded using our platform. SpeakEZ should have access to the user’s microphone if they wish to upload audio using our platform.
3. Audio Analysis: The application should analyze the audio to correctly transcribe the audio into text, in the user’s spoken language.
4. Artificial Intelligence Models: The application should implement AI models in order to produce accurate transcription and translation results.
5. Results Display: the application should show the transcribed audio and translated transcription based on the selected language chosen.

2) Non-functional Requirements

The following are the non-functional requirements for the SpeakEZ application:

1. Usability: The application should be user-friendly; the user should be able to easily navigate the application.
2. Performance: The application should be able to analyze the audio and provide translation results in a timely manner.
3. Security: The application should securely store user information such as login info and the user’s past translations and transcriptions.
4. Compatibility: The application should be accessible on any device, making it universally accessible. Users can record, transcribe, and translate audio seamlessly from their chosen platform (any application that supports a web browser such as a mobile device or computer).

3) Constraints

The following are the constraints for the SpeakEZ application:

1. Technical Constraints: the application should be developed using the appropriate languages, and frameworks, in order to create a responsive and user-friendly web application. The programming language should be limited to JavaScript, with the framework being TailwindCSS.
2. Time Constraint: the application should be developed and tested within 3 months to meet the project deadlines. Atleast the MVP of the application should be completed within this timeframe.
3. Budget Constraint: the application should be developed within a specific budget, where the cost of maintenance be kept to a minimum.

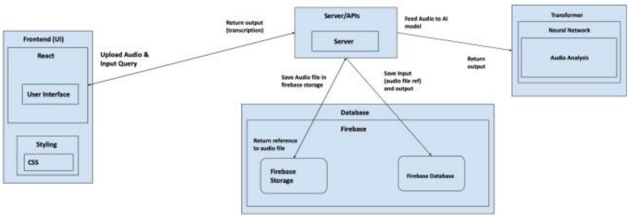


Fig 1. Conceptual Architecture Diagram

The Conceptual Architecture Diagram shown above represents the conceptual software design of our application. It demonstrates how the different components of our application interact with each other to provide a seamless user experience for our users. The user uploads an audio input into our server. The audio is then fed into the AI model. The AI model analyzes our audio and produces output which is then sent back into our server. The user’s results can be securely stored into the firebase database.

IV. DATASET

To provide accurate translation of our transcripted results, we implemented the Helsinki-NLP dataset. Below we detail the dataset’s contributions, limitations, and how it was integrated into our study.

C. Helsinki-NLP Dataset

Our database consists of Helsinki-NLP pre-trained translation models collected from the Language Technology Research Group at the University of Helsinki. The HuggingFace library allows developers to freely access this dataset. At the University, they specialize in “NLP for morphologically rich languages, Cross lingual NLP, and NLP in the humanities” [7].

NLP stand for Natural Language Processing. As stated by IBM in reference [9], NLP is a subfield of Artificial Intelligence and Computer Science- it allows computers to understand the human language in order to communicate with humans using machine learning technology. NLP is very useful; currently it is used for chat boxes or spoken commands to allow software/digital assistants like Siri and Alexa to understand users. There are three different approaches to NLP which include: rule-based NLP, Statistical NLP, and Deep Learning NLP. Rule-based NLP uses decision trees, a machine learning model, which provide results based on specific queries provided. Statistical NLP maps language elements to a statistical result. For example, a language element can be assigned a statistical likelihood, for the meaning to be the equivalent, of the element. Deep learning NLP uses large amounts of unstructured, raw data(voice and text) to produce more accurate results. Some subcategories of this type of NLP which we use in our study are Sequence-to-Sequence(seq2seq) models, and Transformer models. Seq2Seq are mostly used for converting a language into another language precisely. Transformer models are used to break down the language, find its translated equivalents, and merge them back into its grammatical format.

A. Analysis and Integration

We use Helsinki’s database, provided by Huggingface library because it is a free, open source, downloadable dataset of languages with their translated pairs. We found that other competitors such as Google Translate Cloud Translation API, required compensation to use their datasets. Our study would not want to use Google API; we prefer to provide more accurate results to the public and make the application easier for the user to use by allowing audio inputs. Helsinki’s database contains over 100 languages pairs varying a wide array of languages. They also use SentencePiece, which is an unsupervised text tokenizer that automatically generates the sentence into the correct grammatical sentence structure for the targeted language. The model also provides statistics about the language- the number of downloads per language/ its popularity. It provides the source language (the inputted language type) and the target language (the language you would like to translate the word into). It describes the model the database uses for each translation type. For example, if I would like to translate a word from English to Spanish, the source language would be indicated as “en” and the target language would be “sp”. The model for that translation model would be indicated as transformer to let the user know that this model uses an encoder-decoder methods.

For our study, we hardcoded the language selections into our code. The dataset allows us to translate our transcribed English text into any desired chosen language. We also found that for most popular languages such as Spanish, French, and Italian, they were correctly translated from the source language into the targeted language. The models that weren’t popular, Helsinki notify users stating, “the model does not have enough activity to be deployed”. Languages with these notifications, we noticed, contained more errors in the translated results. Another con about this specific dataset, it mostly includes more accurate translations between European languages. It lacks accurate translations between languages of Asian or African backgrounds. We also noticed that the runtime for translation was longer. Each time the user translates a text into a new language, the model downloads the files locally causing the result to take a longer time to execute.

IV. METHODOLOGY

To develop SpeakEZ, we use innovate frameworks to create a responsive and modern web application. We also use a cloud infrastructure and artificial intelligence models to provide a seamless and effective experience for the user. This methodology features the complete system design for our AI Language Transcription and Translation Application.

A. User Frontend

In order to deliver a seamless user experience and ensure the team collaborates effectively, we used a comprehensive set of tools and technology to develop the SpeakEZ application. We used tools such as GitHub, VSCode, and Figma. GitHub allows for the development and collaboration of code. It allows the team to work remotely and still be connected and up-to-date with the code produced. VSCode is used as our chosen code editor. VSCode is flexible and versatile- it supports multiple programming languages and extensions. For brainstorming the UI features of the application, we used Figma to collaborate on the visual aspects. To keep track and prioritize tasks, we used Atlassian Jira software to keep the team organized and focused.

On the technical side, we implemented HTML5, CSS, and JavaScript as the functional programming language- it is the basis for our application. We also implemented React which supports in building dynamic and interactive user interfaces. To create a more modern interface, we implemented libraries such as MaterialUI, and TailwindCSS. Vite was also used to assist with faster development of our front-end application. Firebase is implemented in the backend as our database for securely storing the users’ transcriptions and translations.

By integrating these tools and technologies, SpeakEZ is a reliable and user-friendly platform for enhancing communication between multilingual individuals globally.

B. Machine Learning Model

SpeakEZ implements a whisper model presented by OpenAI. As stated by OpenAI [8], OpenAI’s whisper model is a sophisticated speech recognition system capable of handling multilingual speech recognition, translation, and language identification. As mentioned before, it is trained on a dataset of over 600,000 hours of multilingual data. The model is precise in that it recognizes speech regardless of the accent, colloquial terms used, and background noise presented. [1]

States the whisper model uses an encoder-decoder transformer which breaks down the audio into 30 second chunks and then converts the audio into a log-Mel spectrogram (a spectrogram where the audio frequencies are converted to a mel scale- a unit for pitch [11]). This transformed data is then fed into an encoder where it is processed, and the decoder produces a corresponding text based on the prior trained data from the web. This approach enables the model to effectively perform tasks such as language identification, multilingual transcription, and speech translation. We used OpenAI for our transcription functionality. It allows the user to translate their audio into text, in their spoken language. The model also allows our application to translate the transcribed text into English. Figure 1 demonstrates the architecture of a Sequence-to-sequence model which integrates an encoder-decoder.

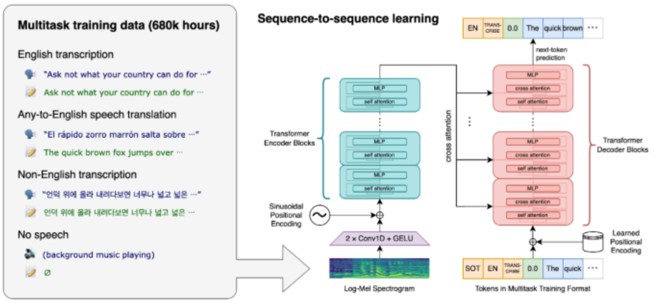

Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ⋯"
Ask not what your country can do for ⋯
Any-to-English speech translation
"El rápido zorro marrón salta sobre ⋯"
The quick brown fox jumps over ⋯
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
⋯
next-token prediction
MLP
self attention
cross attention
Transformer Encoder Blocks
Transformer Decoder Blocks
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
Tokens in Multitask Training Format

Fig 2. Encoder-Decoder Reference

As summarized in [10], an encoder-decoder handles sequential data- it maps given input sequences to contextually appropriate output sequences. The encoder consists of two parts; the self-attention layer, and the feed-forward neutral network. The first layer, the self-attention layer, focuses on words relating to the given input. It looks at the sentence as a whole and compares the importance of each word in the sentence to the other. Once the self-attention compares each word to the other in a sentence, it grants a weight to each word in the sentence depending on the priority. This is needed because it gives meaning to each word and gains context of how the word relates to each other in order for the computer to know exactly what the human/user is trying to say. This improves language understanding for the computer and therefore efficiency in parallel processing of words simultaneously. The feed-forward neutral network then processes the first layer and makes sure it passes the requirements needed for the decoder layer. The feed-forward creates a representation of the words called a hidden state. The hidden state is the result of the token embedded (the weight of each word) and the positional encoding(where each word is in relations to each other in the sentence). The hidden state is sent to the decoder for processing.

The decoder layer contains three layers. It has the first two layers like the encoder- the self-attention layer, and the feed-forward neutral network, and contains the encoder-decoder attention layer. The encoder-decoder attention layer prioritizes the network attention for specific parts of the output for the encoder. As stated in [10], “The decoder uses the positional embeddings to calculate attention scores for each [word]. These attention scores determine to what degree each [word] from the input sequence will affect later [words] therein; in other words, the scores determine how much weight each [word] has in other [words’] determinations when generating output sequences”. The decoder makes sure that the output aligns with the input’s meaning and context. The encoder-decoder’s ultimate goal is to mimic the understanding of how humans process and analyze information. The layers for both the encoder-decoder are shown in Figure 2.

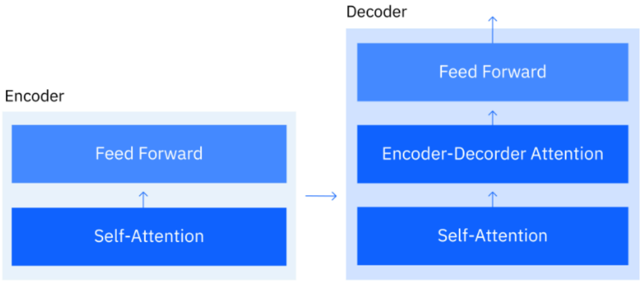


Fig 3. Encoder-Decoder Layers

C. Firebase Backend API

As addressed in [4], Firebase is built on Google Cloud infrastructure in which developers use the service to store user generated content. In firebase users are able to store images, audio, videos, and other forms of content. For the SpeakEZ application, we will be deploying firebase in the backend to store the user’s transcribed and translated work. Firebase will also be used for user login authentication. Firebase is reliable for its security, scalability, and robustness. Deploying firebase on the cloud allows for real-time data synchronization across devices, ease of front-end development due to its accessible features, offline data persistence, and saves on compute time.

D. Workflow

The workflow of SpeakEZ involves the user first inputting audio, whether it be imported from their device or recorded using our application. That audio is then transcribed to text in the user’s spoken language with the help of the OpenAI API. The application then uses the API to translate the transcribed text into English. Users are then able to translate the text into their targeted language with the assistance of the Helsinki-NLP model. This workflow allows for the accurate and real time transcription of translation of audio for users, making communication between users of different languages simple and effortless.

E. Deployment

The SpeakEZ application is fully deployed on the Google Cloud Platform. The Google Cloud Platform allows SpeakEZ to be deployed in a secure, scalable, and highly accessible manner for the user. The React based web application which integrates TailwindCSS and MaterialUI serves as the responsive frontend. The Firebase database serves as file storage for user translation and transcription results. The Firebase also manages user authentication for their login experience. Hosting SpeakEZ on Google Cloud allows for the hosting of the back-end application in a virtual environment. This is beneficial because it solves the issue of translating causing a longer runtime. As mentioned before, when a user tries to translate their text, the model downloads the files locally causing the result to take a longer time to execute. Hosting the backend virtually on the Cloud platform allows for faster compute time when translating and removes the option of having the user's machine to be slowed down because it uses too much of the user's machine resources.

V. PRODUCT RESULTS

A widely used metric for evaluating transcription accuracy is the Word Error Rate (WER). The WER calculates the proportion of errors in a transcribed text compared to a reference text. [5] breaks down the WER score of the OpenAI Whisper and Google's Speech-to-Text which we will break down now. OpenAI's Whisper model achieves a median WER of 8.06%, demonstrating a significantly higher accuracy compared to Google Translate's WER, which ranges from 16.51% to 20.63%. This indicates that Whisper produces fewer errors in transcription, making it more reliable for accurate audio processing. As a result, incorporating Whisper into our application enhances its overall accuracy and trustworthiness, ensuring a better user experience.

As for the user experience, SpeakEZ provides a login page for users to sign up for our application. The signup will allow users to securely store their transcriptions, and translations for future use. The user will be able to upload a .mp3 file and or record an audio clip using our application. The user will be able to transcribe their audio, and then if desires, translate their transcription into their desired language.

These features will provide the user with an efficient and seamless user experience. Figures 4 to 10 demonstrate the features mentioned above.

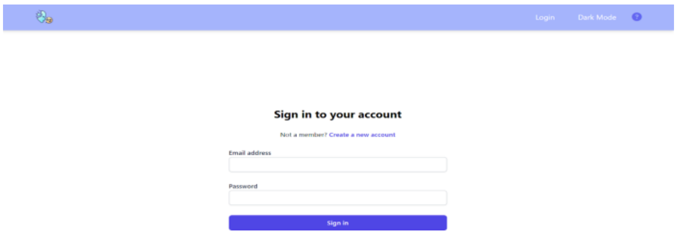


Fig 4. Home Login Page for User

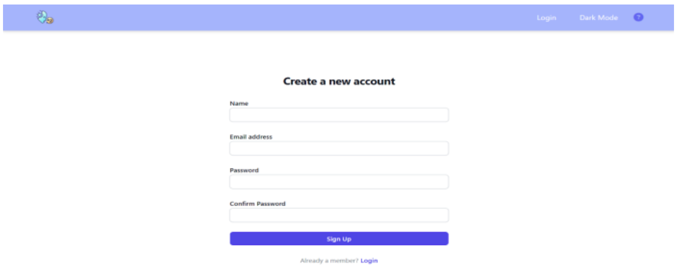
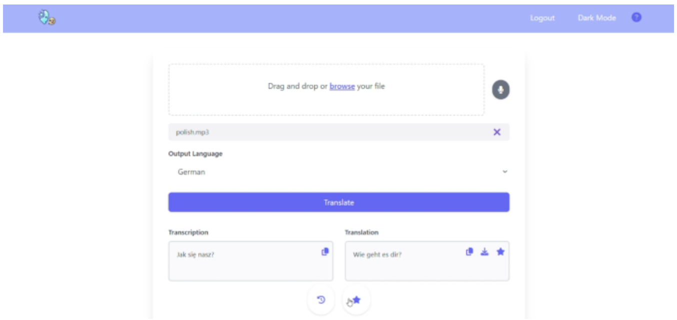


Fig 5. Sign-up Page for User



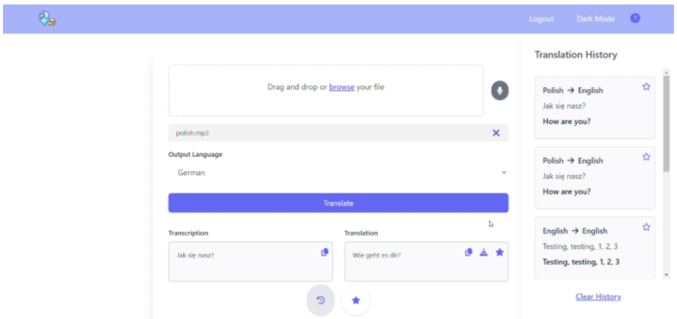
polish.mp3

German

Jak się nasz?

Wie geht es dir?

Fig 6. Transcription & Translation Page



polish.mp3

German

Jak się nasz?

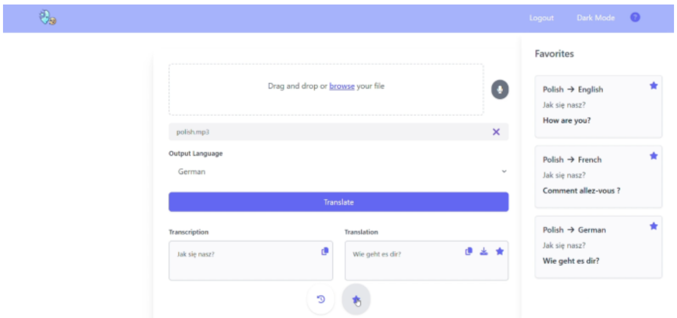
Wie geht es dir?

Polish → English
Jak się nasz?
How are you?

Polish → English
Jak się nasz?
How are you?

English → English
Testing, testing, 1, 2, 3
Testing, testing, 1, 2, 3

Fig 7. History Section



polish.mp3

German

Jak się nasz?

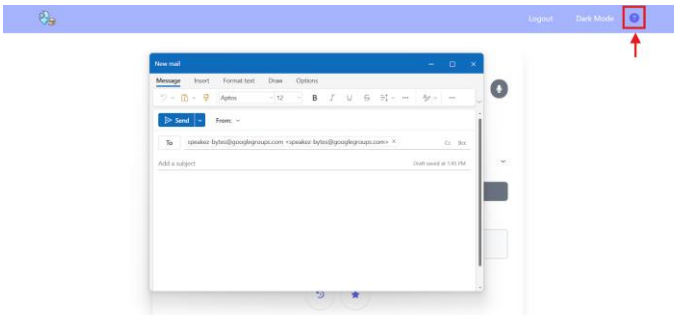
Wie geht es dir?

Polish → English
Jak się nasz?
How are you?

Polish → French
Jak się nasz?
Comment allez-vous ?

Polish → German
Jak się nasz?
Wie geht es dir?

Fig 8. Favorites Section



speakez-bytes@googlegroups.com <speakez-bytes@googlegroups.com>

Draft saved at 1:45 PM

Fig 9. Contact Team Icon

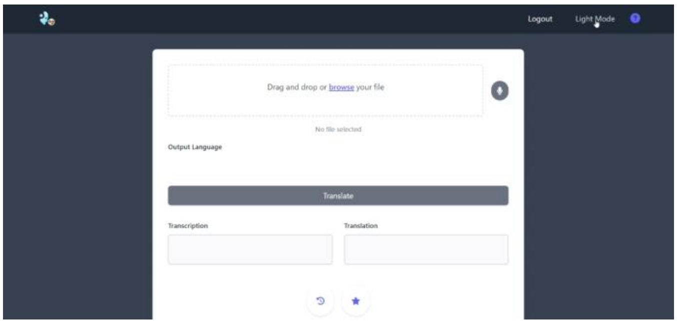


Fig 10. Changing Theme (Dark Mode)

VI. CONCLUSION

In conclusion, SpeakEZ is a translator web application that processes audio input from a file or recorder, converts it into a transcribed text, and translates it into a specified language. By enhancing communication and bridging language barriers, it aims to bring the world closer through AI.

Future development of the project can involve seeking advice from professors and incorporating customer feedback to create more user stories, making the application more seamless and intuitive. Additional work could include adding functionalities like supporting larger file sizes for uploads, enabling live translation while receiving audio input, and allowing users to upload reports, technical conference papers, or documents for direct translation into multiple languages. Another feature to consider is adding timestamp text while transcribing for better organization. Furthermore, the implementation of voice volume controls would also make it easier for those who rely on auditory outputs for clarity or have visual impairments. The dual-option output would enhance accessibility, catering to those of diverse user needs.

There is significant potential to optimize the performance of the application, making it more efficient. SpeakEZ has the potential to develop into a marketable product. With recent advancements in AI models and frameworks, we can leverage this technology to make a positive impact on the world.

REFERENCES

[1] A. Radford, J. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” Dec. 2022. Available: https://cdn.openai.com/papers/whisper.pdf

[2] Chenna, “Bayes Theorem - Proof, Formula and Solved Examples,” BYJUS, 2019. https://byjus.com/maths/bayes-theorem/

[3] F. Chessa and G. Brelstaff, “Going beyond Google Translate?,” P-arch (Science and Technology Park of Sardinia), vol. 1, pp. 108–113, Sep. 2011, doi: https://doi.org/10.1145/2037296.2037324.

[4] Firebase, “Cloud Storage | Firebase,” *Firebase*, 2019. https://firebase.google.com/docs/storage

[5] “Gladia - OpenAI Whisper vs Google Speech-to-Text vs Amazon Transcribe: The ASR Rundown,” *Gladia.io*, 2024. https://www.gladia.io/blog/openai-whisper-vs-google-speech-to-text-vs-amazon-transcribe

[6] Google Translate — Remotely Nomad, “Remotely Nomad,” Remotely Nomad, 2014. https://remotelynomad.com/shop/p/google-translate (accessed Nov. 19, 2024).

[7] “Helsinki-NLP (Language Technology Research Group at the University of Helsinki),” huggingface.co. https://huggingface.co/Helsinki-NLP

[8] “Introducing Whisper,” Openai.com, 2022. https://openai.com/index/whisper/

[9] J. Holdsworth and C. Stryker, “What Is Natural Language Processing?,” IBM, Aug. 11, 2024. https://www.ibm.com/topics/natural-language-processing

[10] J. M. Ph.D and J. Noble, “Encoder-Decoder Model,” Ibm.com, Dec. 02, 2024. https://www.ibm.com/think/topics/encoder-decoder-model (accessed Dec. 08, 2024).

[11] L. Roberts, “Understanding the Mel Spectrogram,” Medium, Mar. 14, 2020. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

[12] P. Stapleton and B. Leung Ka Kin, “Assessing the accuracy and teachers’ impressions of Google Translate: A study of primary L2 writers in Hong Kong,” English for Specific Purposes, vol. 56, pp. 18–34, Oct. 2019, doi: https://doi.org/10.1016/j.esp.2019.07.001.