A Major Project Report on

WHISPERMAIL

Submitted by
Vedant Chandak (Roll No: BT4104)
Tushar Katore (Roll No: BT4115)
Vaibhav Rathod (Roll No: BT4126)

Under the guidance of
Dr. S. R. Chaudhary

In partial fulfillment of the award of Bachelor of Technology in Computer Science and Engineering

Department of Computer Science and Engineering
Maharashtra Institute of Technology(An Autonomous),
Chh. Sambhajinagar (Aurangabad)
[2024-25]

DECLARATION

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included; we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any data/fact/source in my submission. we understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Chh. Sambhajinagar (Aurangabad).
Date:

Signature and
Name of students:
Vedant Chandak
Tushar Katore
Vaibhav Rathod

CERTIFICATE

This is to certify that the Major Project report entitled "**WHISPERMAIL"**, submitted by **Vedant Chandak (BT4104), Tushar Katore (BT4115), Vaibhav Rathod (BT4126)** is the bonafied work completed under my supervision and guidance in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering, Maharashtra Institute of Technology under Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar (Aurangabad).

Place: Chh. Sambhajinagar (Aurangabad).
Date:

Dr. S. R. Chaudhary
Guide

Prof. Dr. S. L. Kasar
Head of Department

Dr. N. G. Patil
Director
Maharashtra Institute of Technology
Chh. Sambhajinagar (Aurangabad).

APPROVAL CERTIFICATE

This Major Project report entitled "**WHISPERMAIL"** by **Vedant Chandak (BT4104), Tushar Katore (BT4115), Vaibhav Rathod (BT4126)** is approved for Bachelor of Technology in Computer Science and Engineering, Maharashtra Institute of Technology under Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar (Aurangabad).

Place: Chh. Sambhajinagar (Aurangabad).
Date:

Examiner: _____
(Signature)

_____
(Name)

INDEX

	Title	Page no.
	Declaration	
	Certificate	
	Acknowledgement	
	Approval certificate	
	List of Figures	i
	List of Tables	ii
	Abstract	iii
1.	**INTRODUCTION**	**1**
	1.1 Necessity	2
	1.2 Problem definition	2
	1.3 Objectives	2
	1.4 Scope and limitations	3
	1.5 Applications	4
2.	**LITERATURE SURVEY**	**5**
	2.1 Literature	5
	2.2 Algorithm	9
3.	**SYSTEM DESIGN AND DEVELOPMENT**	**15**
	3.1 API & Web-scrapping	15
	3.2 Google Packages and API	16
	3.2.1 Google packages	16
	3.2.2 Google API	16
	3.3 Project Design	17
	3.3.1 Core Feature and Functional Components	17
	3.3.1 Technical Approach and Architecture	19
	3.4 Project Methodology	25
4.	**PERFORMANCE ANALYSIS**	**33**
5.	**CONCLUSION**	**35**
	REFERENCE	

List of Figures

Figure	Illustration	Page
1.1	Audio in wave form	1
2.1	HMM Diagram	9
2.2	CNN Diagram	11
2.3	RNN Diagram	11
2.4	GMM Diagram	12
3.1	Google Translate	16
3.2	Input and Output language selection	20
3.3	Sender details and Mail subject	21
3.4	Mail body	22
3.5	Add attachments	23
3.6	Receivers mail id	24
3.7	SDLC Diagram	25
3.8	Flow Diagram of Tech stack	27
3.9	Prototype code 1	29
3.10	Prototype code 2	30
3.11	Prototype 1	31
3.12	Prototype 2	32
4.1	Use Case Diagram	33

List of Tables

Table No.	Illustration	Page
2.1	Literature Table	6
4.1	Table of Test Cases	34

ABSTRACT

WhisperMail is an innovative web app that combines speech-to-text translation with integrated email functionality to streamline communication and accessibility. By using advanced speech recognition, WhisperMail allows users to compose and send emails by converting spoken words directly into text. This feature is especially beneficial for individuals with hearing impairments or limited typing abilities, as well as for users seeking hands-free communication. Supporting multiple languages and customizable settings, the app enables cross-language email interactions and adapts to diverse user needs.

WhisperMail's seamless integration with web platforms ensures that emails can be sent across various devices, maintaining accessibility anywhere. Privacy and data security are prioritized, with stringent measures to protect users' speech data and email content. The app also offers additional tools, such as playback and text editing, to review and refine messages before sending. Through regular improvements and a focus on user feedback, WhisperMail is transforming digital communication, fostering inclusivity, and enhancing productivity in today's connected world.

CHAPTER 1
INTRODUCTION

WhisperMail combines the power of speech-to-text technology with mail integration to make communication effortless and hands-free. It allows users to convert their spoken words into text with high accuracy and then seamlessly compose and send emails. By using advanced algorithms, WhisperMail ensures precise transcription, even for different languages. WhisperMail is designed to enhance productivity, especially for people looking for accessibility solutions or hands-free communication. Whether it's for personal use or professional tasks, WhisperMail is transforming how we interact with technology, making email composition faster, easier, and more inclusive.

What is Sound?

Sound is defined as vibrations that travel through the air or another medium as an audible mechanical wave. It is produced from a vibrating body. The vibrating body causes the medium (water, air, etc.) around it to vibrate thus producing sound. The sound is produced when something vibrates. Sound waves consist of areas of high and low pressure called compressions and rarefactions, respectively. Vibrations that travel through the air or another medium, can be heard when they reach a person's or animal's ear.

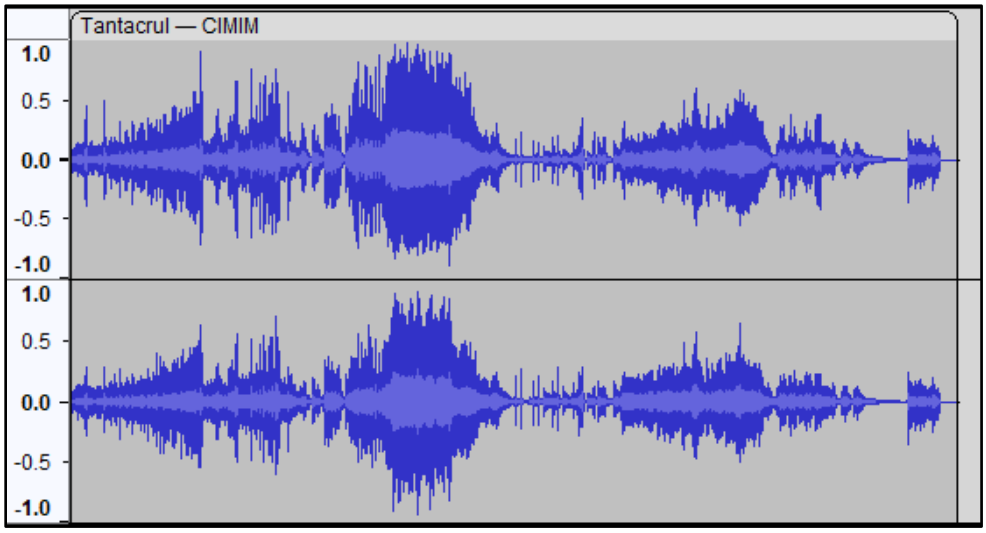

Figure 1.1 Audio in wave form

1.1 Necessity

Speech-to-text translation is required because effective communication is required in a variety of situations. It makes things easier for people with impairments to access, allows for real-time transcription of lectures and meetings, improves user experience with voice-activated devices, and breaks down linguistic barriers to support multilingual communication. Furthermore, reliable transcription of spoken dialogue is essential for documentation and analysis in domains like healthcare and legal services. In general, speech-to-text translation makes information more accessible and interaction more inclusive by meeting the need for seamless communication across many platforms and contexts.

1.2 Problem Definition

The difficulty of accurately and quickly translating spoken language into written text is the issue that speech-to-text translation attempts to solve. To get accurate transcriptions, obstacles including background noise, accent variances, and language complexity must be overcome. The objective is to create systems that can accurately comprehend spoken words in a variety of languages and settings and transcribe them. Important factors to take into account include also making sure that interfaces are user-friendly and accessible to people with disabilities. Thus, the creation of reliable speech-to-text translation technologies that satisfy the needs of various consumers and applications is included in the problem definition.

1.3 Objectives

- To facilitate real-time transcription of spoken content for increased productivity.
- To provide seamless cross-language communication by supporting multiple languages.
- To ensure secure handling of users mail data and personal information.
- Provide a solution for individuals with disabilities or those who prefer hands-free communication, improving inclusivity and usability.

1.4 Scope and Limitations

- **Scope:**
  - Support for a broad variety of languages and dialects is necessary to guarantee inclusivity and accessibility for people everywhere.
  - Application integration is the process of incorporating speech-to-text translation features into different platforms and applications, such as communication tools, transcription software.
  - Real-time transcription is the use of features for live events, meetings, lectures, and discussions that allow spoken words to be instantly converted into text.

- **Limitations:**
  - Limitations on Accuracy: Not withstanding its progress, speech-to-text systems may still have trouble accurately transcribing specific accents, dialects, or languages, which could result in inaccuracies in the text that is produced.
  - Ambient Noise: In noisy or congested circumstances, background noise can significantly affect the accuracy of speech recognition.
  - Complex Speech Patterns: Rapid speech, overlapping conversations, and complex speech patterns can be difficult for voice-to-text systems to translate, which can result in inaccurate transcriptions.
  - Vocabulary Restrictions: Speech-to-text systems may not be able to accurately transcribe texts if they have trouble identifying slang, specialised terminology, or rare words.
  - Users may be concerned about the security and privacy of the information they say, especially if it is being recorded, saved, or sent via networks.
  - Dependent on the clarity of the user's speech for optimal performance. Requires internet connectivity for mail delivery and certain advanced features.

1.5 Applications

- **Accessibility**:
    - Speech-to-text translation can make digital content more accessible to individuals with physical disabilities. It allows them to engage with spoken content by converting it into text in real-time.
- **Language Learning**:
    - Speech-to-text translation can aid language learners by providing real-time transcription of spoken language, helping them improve their listening comprehension and pronunciation.
- **Voice Typing**:
    - Instead of typing on a keyboard, users can dictate text using speech-to-text translation in word processing software, email clients, or messaging apps.
- **Language Translation**:
    - Speech-to-text translation can be combined with machine translation techniques to create systems that convert spoken input in one language into text in another language.
    - This is useful for multilingual communication and international business.
- **Text after speech:**
    - User can copy/cut the text which converted after speech. It will save time of typing for the users.
- **Educational and Personal Use:**
    - Assist students and educators in quickly transcribing ideas, notes, or assignments and emailing them directly.
    - Simplify personal communication for users who prefer speaking over typing.

CHAPTER 2
LITERATURE SURVEY

2.1 Natural Language Processing(NLP):

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant. NLP plays a crucial role in bridging the gap between human communication and computer understanding, enabling machines to process and interact with human language data in various forms, such as text, speech, and even gestures.

- Speech processing – field that covers speech recognition, text-to-speech and related tasks.
- Statistical natural-language processing –
    - Statistical semantics – A subfield of computational semantics that establishes semantic relations between words to examine their contexts.
    - Distributional semantics – A subfield of statistical semantics that examines the semantic relationship of words across a corpora or in large samples of data.

❖ **Phases of WhisperMail:**

1) **Input Phase**:
    - User Inputs: Users enter their email details, including sender email, password, recipient email, subject, and body text. They can also upload an attachment.
    - Frontend Form: The input phase is facilitated through a web-based form created using HTML, CSS, and JavaScript.
2) **Pre-processing**:
    - Form Submission: The JavaScript code captures form data and sends it to the Flask backend using the fetch API.

- Data Validation: The backend processes and validates the data, ensuring that required fields are filled and the attachment is correctly included.
- Email Composition: The Flask backend prepares the email using libraries such as smtplib or flask-mail.
- SMTP Authentication: The backend connects and authenticates with the SMTP server (e.g., Gmail SMTP) to send the email.

3) **Output Phase:**
   - Email Sending: The backend sends the email to the recipient using the SMTP connection.
   - Response Generation: The Flask backend returns a JSON response indicating whether the email was sent successfully or if there was an error.
   - User Notification: The JavaScript code on the frontend displays an alert with the status of the operation (success or failure).

Table 2.1 Literature

Aspects	Description	Method	Source
Focus	Reviews research on speech recognition (automatic speech-to-text conversion) with a focus on its application in translation systems.	Analyzes existing research papers on Automatic Speech Recognition (ASR), Machine Translation (MT), and how they are combined for speech-to-text translation.	A Literature Survey: Spoken Language Translation CFILT – IITB [1]
Target Audience	Researchers in machine translation and speech recognition interested in combined technologies.	Primarily focuses on research papers that explore how ASR and MT techniques are	Expressive body capture: 3D hands, face, and body from a single image

		integrated for speech-to-text translation.	(Pavlakos et al., 2019) [2]
Applications & Users	Speech-to-text translation benefits real-time communication, education (e.g., for students with disabilities), and generating text from lectures/meetings.	Review studies on the use of speech-to-text translation in various applications. Consider research on educational technology or accessibility tools.	A Review of the Literature on Computerized Speech-to-Text Accommodations. [3]
Combined technology	Speech-to-text translation integrates Automatic Speech Recognition (ASR) and Machine Translation (MT). ASR converts speech to text, and MT translates the text to another language.	Analyze research papers on ASR techniques (e.g., Hidden Markov Models, Deep Learning) and MT approaches (e.g., Statistical MT, Neural MT) to understand how they are combined for speech-to-text translation.	A Literature Survey: Spoken Language Translation (CFILT - IITB) [1]
Accuracy & Challenges	Speech-to-text translation accuracy depends on ASR and MT performance. Challenges include noise, accents, and idiomatic expressions.	Review research on methods for improving speech recognition accuracy (e.g., noise reduction techniques) and MT effectiveness (e.g., using large language models, incorporating context).	A recent survey paper on Text-To-Speech Systems (International Journal of Advanced Research in Science and Technology) [5]
Evaluation metrices	Speech-to-text translation systems are evaluated using metrics like Word Error	Analyze research papers on evaluation metrics used for speech-to-text	A Review of Speech Recognition Technologies (IEEE

	Rate (WER) for ASR and BLEU score for MT.	translation systems, including WER for ASR and BLEU score or alternative metrics for MT (e.g., human evaluation).	Transactions on Speech and Audio Processing) [5]
Future Directions	Research areas include improving accuracy for specific languages/environments, personalizing translation based on user context, and reducing latency (real-time translation).	Identify research trends and open problems in reviewed papers. Explore recent advancements in ASR, MT, and related fields (e.g., speaker diarization).	Latest research papers and conference proceedings on ASR, MT, and speech-to-text translation. [5]
Mail Functionality	Enables users to send translated output as email content directly from the application.	Integrated Python's SMTP library for mail services.	Documentation on Python's SMTP module.
Reusability	Ensures functions like 'speak' and 'stop' can be invoked multiple times in a session without restarting the application.	Modular function development approach.	Python scripting and Flask.
Development Approach	Prototyping methodology in the Software Development Life Cycle (SDLC).	Iterative design and feedback loops for implementing and testing features.	Standard SDLC practices.
UI Technologies Used	Web-based interface built for smooth user interaction with the tool.	HTML, CSS, JavaScript for the front end; Flask for routing and interaction.	Open-source UI design resources. [8] [9]

2.2 Algorithms

Speech-to-text (STT) translation systems employ a variety of methods, each with unique advantages and disadvantages. Typical algorithms include the following:

1) Hidden Markov Models (HMMs):

- HMMs are statistical models that assume the system being modeled is a Markov process with hidden states. This means the system transitions from one state to another with certain probabilities, but these states (e.g., words or phonemes in speech) are not directly observable. Instead, what we observe are outcomes (e.g., acoustic features).
- In WhisperMail, HMMs simulate the connection between spoken words (hidden states) and the acoustic signals (observed features) extracted from the speech. For example, if a user says "Hello," the system maps the acoustic features of this utterance to the hidden phoneme sequence that forms the word "Hello."

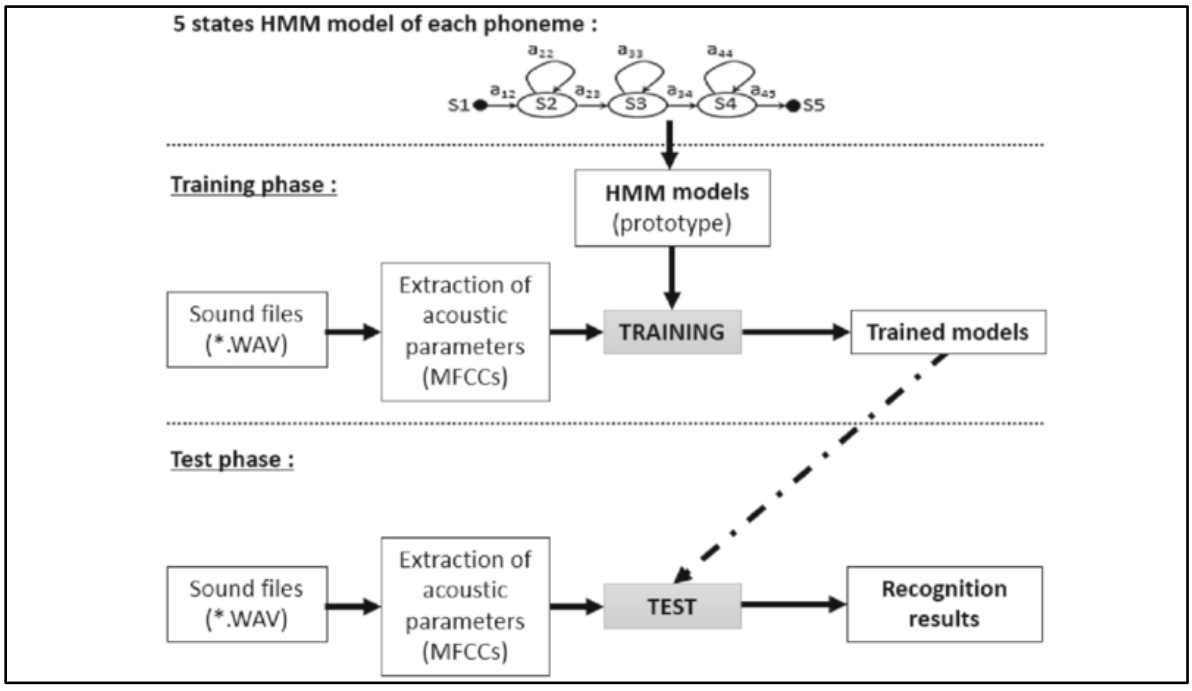

Figure 2.1 HMM Diagram

- **Key Components of HMM:**

- States: The system comprises a set of hidden states, which are not directly observable but influence the observable outcomes.
- Observations: Each hidden state generates a probability distribution over observable outcome. These observations are visible to the observer.
- Transition Probabilities: HMM assumes that transitions between hidden states follow a Markov process, where the probability of transitioning to a particular state depends only on the current state.
- Emission Probabilities: Each hidden state emits observable outcomes with associated probabilities.

➢ **Applications of HMM:**

- Speech Recognition: HMMs are used to model phonemes and acoustic features in speech recognition systems.
- Natural Language Processing: In NLP, HMMs are utilized for tasks such as part-of-speech tagging and named entity recognition.
- Bioinformatics: HMMs are employed for sequence analysis, including gene prediction, protein structure prediction, and alignment of DNA sequences.
- Finance: HMMs are applied in financial modeling for tasks such as predicting stock prices, identifying market regimes, and analyzing time series data.

➢ **Advantages of HMM:**

- Flexibility: HMMs can model complex sequential data with a relatively simple framework.
- Probabilistic Framework: HMMs provide a probabilistic framework for modeling uncertainty, allowing for robust inference.
- Efficient Inference: Efficient algorithms such as the Viterbi algorithm and the forward-backward algorithm enable fast inference of hidden states from observed data.

2) Convolutional Neural Networks (CNN):

- CNNs are deep learning models designed to process grid-like data, such as images or spectrograms (a visual representation of sound). They excel at identifying patterns through convolutional layers that focus on local features in the input data.
- WhisperMail can use **CNNs** to analyze spectrograms of speech signals. By doing so, CNNs can extract detailed acoustic features such as phonetic patterns or variations in pitch and tone that contribute to accurate transcription.

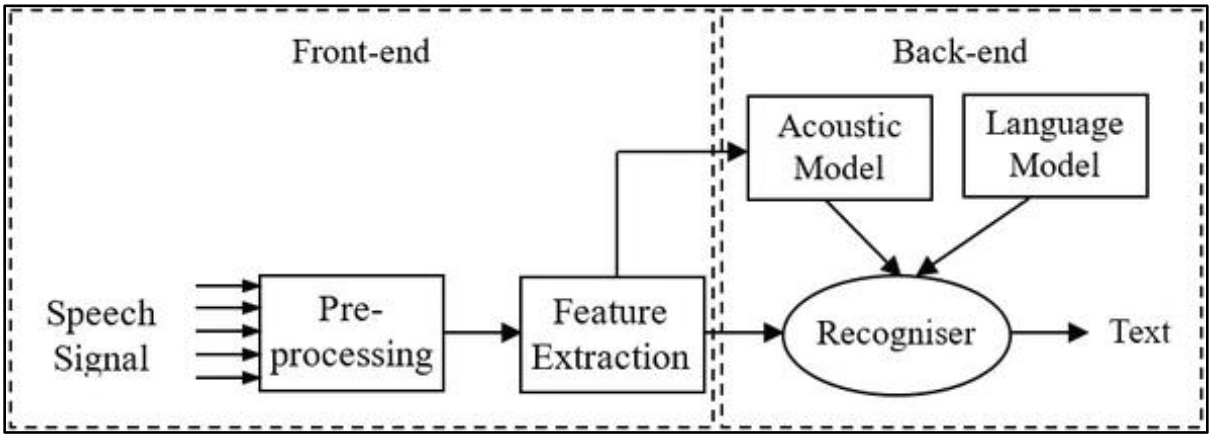

Figure 2.2 CNN Diagram

3) Recurrent Neural Networks (RNNs):

- RNNs are a type of neural network specifically designed for sequential data, like speech. Unlike traditional networks, RNNs have a "memory" that captures information from previous inputs, making them ideal for recognizing patterns over time.

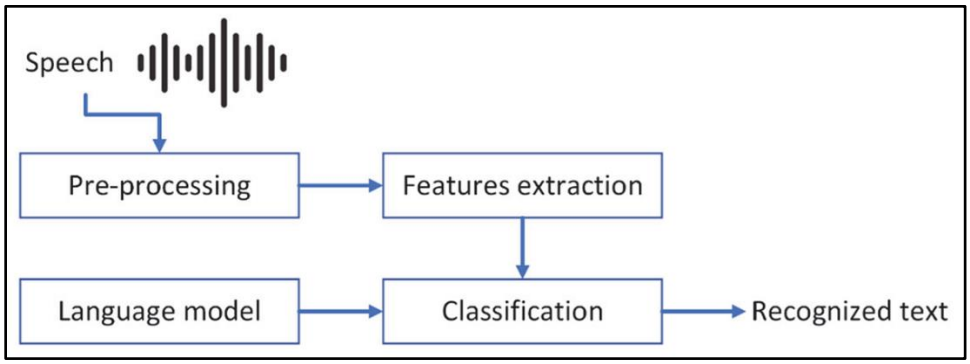

Figure 2.3 RNN Diagram

4) Gaussian Mixture Model (GMM):

- GMMs are probabilistic models that assume the observed data is generated from a mixture of several Gaussian distributions. Each distribution represents a specific component (e.g., phonemes in speech).WhisperMail can use GMMs to cluster and classify acoustic features into phonemes or words.

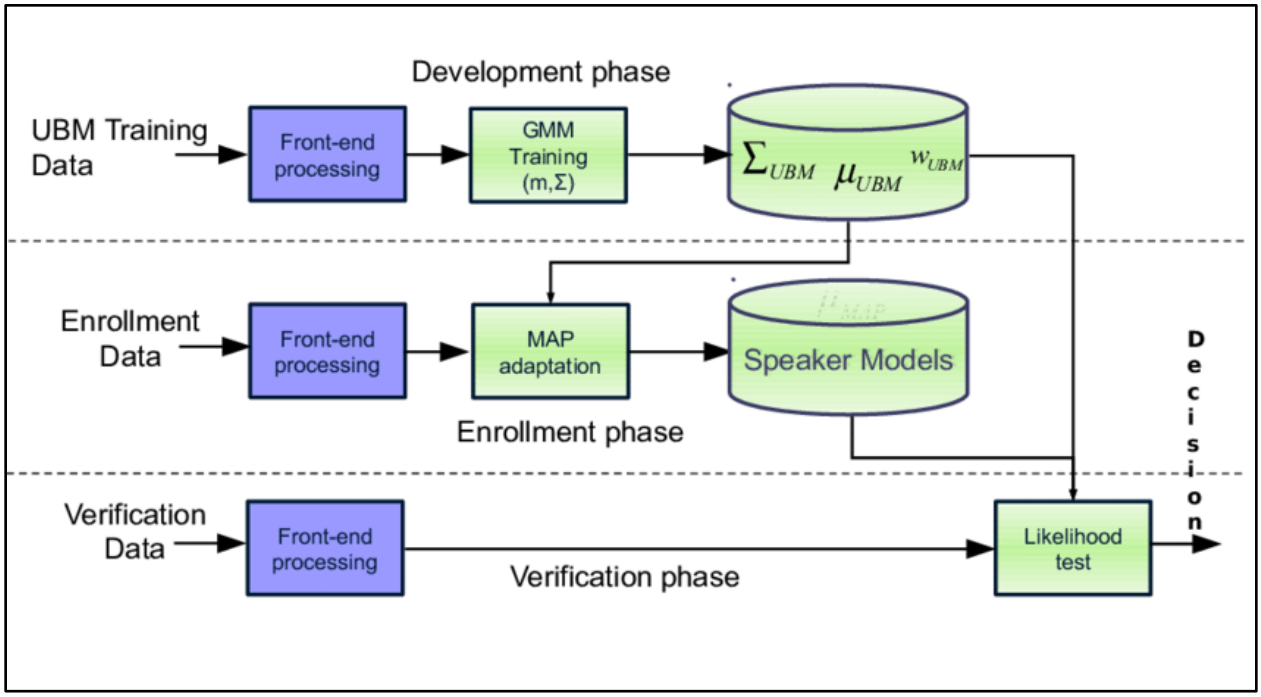

Figure 2.4 GMM Diagram

➢ **Key Components of GMM:**

- Components: GMM assumes that the observed data is generated from a mixture of several Gaussian distributions, each characterized by its mean and covariance.
- Weights: Each component in the mixture is associated with a weight, representing the probability of selecting that component when generating data.
- Probability Density Function (PDF): The probability density function of a GMM is the weighted sum of the individual Gaussian distributions, providing a flexible model for capturing complex data distributions.

- **Applications of GMM:**
    - Clustering: GMM can be used for clustering data points into groups based on their probability densities, with each cluster represented by a Gaussian component.
    - Density Estimation: GMM can estimate the underlying probability density function of the observed data, enabling analysis of data distribution and anomaly detection.
    - Data Generation: GMM can generate new data points that resemble the distribution of the observed data, making it useful for data synthesis and augmentation.

- **Advantages of GMM:**
    - Flexibility: GMM provides a flexible framework for modeling complex data distributions that may not be adequately captured by a single Gaussian distribution.
    - Scalability: GMM scales well to large datasets and can handle high-dimensional data efficiently.
    - Probabilistic Interpretation: GMM provides a probabilistic interpretation of the data, allowing for uncertainty quantification and principled decision-making.

A. Accuracy Check Metrics:

- Word Error Rate (WER):
    - Word Error Rate is the most widely used metric for evaluating the accuracy of STT models. It measures the percentage of words that are incorrectly predicted.

- Character Error Rate (CER):
    - Character Error Rate is similar to WER but operates at the character level instead of the word level. This can be particularly useful for languages or applications where word segmentation is challenging.
- Accuracy:

- Accuracy measures the percentage of correctly transcribed words over the total number of words. It is often used in conjunction with WER to provide a more intuitive sense of performance.

CHAPTER 3

SYSTEM DESIGN AND DEVELOPMENT

3.1 API & Web-scrapping:

Web scraping is the automated process of extracting large amounts of data from websites quickly and efficiently. It involves parsing the HTML structure of web pages to retrieve specific information and store it in a structured format for analysis or further use. This technique is invaluable for various purposes such as market research, competitive analysis, and data-driven decision making.

- **Key Components of Web Scraping:**
  - **Request:** The process begins with sending an HTTP request to the targeted website to retrieve its HTML content.
  - **Parsing:** Once the HTML content is obtained, web scraping tools parse through the markup language to identify and extract the desired data elements.
  - **Data Extraction:** Using techniques like XPath, CSS selectors, or regular expressions, relevant data such as text, images, or links are extracted from the HTML structure.
  - **Data Storage:** Extracted data is then structured and stored in a format suitable for further analysis or integration into databases or applications.

- **Considerations and Challenges:**
  - Ethical Concerns: Web scraping should be conducted ethically and in compliance with the website's terms of service to avoid legal issues.
  - Dynamic Content: Websites with dynamic content loaded via JavaScript may require more advanced scraping techniques or tools capable of rendering JavaScript.
  - Robustness: Scraping scripts should be robust enough to handle changes in website layout or structure without breaking.

- **Benefits:**

- Data Accessibility: Web scraping provides access to vast amounts of data from the internet, enabling businesses and researchers to gather valuable insights.
- Automation: Automation of data extraction tasks saves time and resources compared to manual data collection methods.
- Competitive Advantage: By monitoring competitors' websites, businesses can gather intelligence on pricing, product offerings, and market trends to gain a competitive edge.

3.2 Google packages and API :

3.2.1 Google packages

- gTTS (Google text to speech)
    - Stands for Google Text-to-Speech.
    - Converts text into speech and generates audio files.
    - Example Use: Reading the transcribed text aloud.

3.2.2 Google API

- Free for first 500000 characters in a month
- High Accuracy
- Has multiple Indian languages like Hindi, Marathi, Bengali and many more

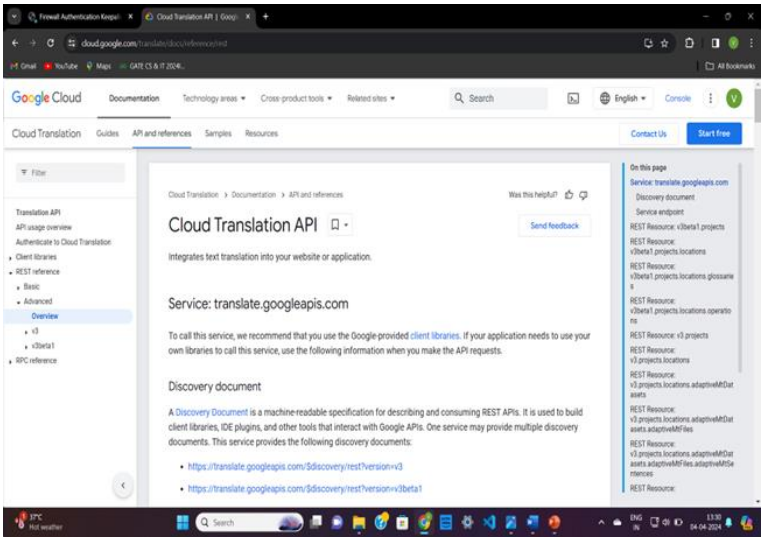

Figure 3.1 Google Translate

3.3 Project Design

- The Speech-to-Text Email Composer is a cutting-edge tool designed to simplify and speed up email composition by enabling users to dictate emails using voice commands. With the integration of advanced speech recognition, this project allows users to create email subjects and bodies simply by speaking, making the process more intuitive, hands-free, and efficient.
- This application is ideal for users who are on the go, have accessibility needs, or simply prefer the convenience of speaking over typing. By combining the power of speech recognition with email integration, it also eliminates manual errors, provides a seamless workflow for email drafting, and significantly enhances productivity.

3.3.1 Core Features and Functional Components

- **Speech-to-Text Translation**
  - **Real-Time Voice Recognition**:
    - This component allows users to dictate text that is then transcribed in real time, enabling them to see their words on the screen immediately.
  - **Error Handling**:
    - By offering a review option, users can go through the transcribed text to ensure accuracy before sending the email.
  - **Multi-Language Support**:
    - Users can speak in various languages (if supported by the speech recognition API), expanding the app's usability for non-English speakers.
  - **Auto-Punctuation and Formatting**:
    - For a more polished result, the system may automatically format the transcription, such as capitalizing the start of sentences and adding punctuation.

- **Email Integration**
  - **Voice-Driven Email Composition**:
    - Users can dictate the Subject and Body of the email separately, with easy commands to switch between the two. For example, the app might have commands like “subject start” or “body start” to capture specific email sections else user can directly start to speak.

  - **Email Sending API Integration**:
    - The application integrates with an email API (like SMTP, Gmail API, or SendGrid), enabling it to send the transcribed message directly from the app.
  - **Attachment Support**:
    - Depending on the app’s configuration, users could attach files to emails via voice commands or pre-defined options, enhancing functionality.

- **User Interface and Experience Design**
  - **User-Friendly GUI**:
    - A clean, interactive layout designed to keep the process simple and clear, with **start** and **stop** buttons for speech input.
  - **Easy-to-Follow Prompts**:
    - Clear prompts guide users on how to proceed at each step (e.g., “Say your subject now”, “Now say your message”), making it suitable for users of all technical levels.

- **Automation and Accessibility**
- **Enhanced Accessibility**: The application provides a more accessible solution for email composition, catering to a wider user base, including those who find typing challenging or inconvenient.
- **Text Correction and Grammar Check**: An integrated grammar check could help ensure that the transcribed text is polished before sending.

3.3.2 Technical Approach and Architecture

- **Frontend Development**
  - **Technology Stack**: The frontend is developed using HTML, CSS, JavaScript.
  - **Interactive UI/UX Design**: The frontend design includes clear visual elements, interactive buttons, and feedback indicators to make it intuitive and responsive.

- **Backend Integration**
  - **Speech Recognition API**:
    - **Google Speech-to-Text**, or similar could be integrated to process the voice input into text.
    - **Noise Filtering**: Configuring the API to handle background noise or adjust sensitivity can improve transcription accuracy.
  - **Email API**:
    - Integration with an email API (e.g., SMTP, Gmail API, or SendGrid) allows the app to send emails directly from the user interface.
    - **Security**: Ensure user email accounts and personal data are securely handled, especially during email transmission.

- **Benefits and Use Cases**
  - **Enhanced Productivity**:
    - By enabling users to compose emails by speaking, this tool reduces the time spent on manual typing, making it particularly useful for busy professionals, students, and people managing high email volumes.
  - **Increased Accuracy**:
    - The app's speech-to-text technology, combined with review options, minimizes typographical errors and ensures clarity in communication, resulting in well-structured and error-free emails.
  - **Improved Accessibility**:

- For users with limited mobility, visual impairments, or those who find typing difficult, this app provides a voice-driven, accessible alternative to traditional email composition methods.

The Voice-Driven Email Composer is a speech-to-text application with email integration, enabling users to compose emails by speaking. It transcribes spoken words into text for the email's subject and body, making email drafting faster and hands-free. Designed with an intuitive interface, users can start, pause, and stop recording, review, and edit transcribed text, ensuring accuracy.

❖ **How to use WhisperMail:**

Figure 3.2 Input and Output language selection

1. **Select input and output language as per your choice.**
    - The first step is to select your desired input and output languages.
        - **Input Language**: This is the language you will speak in for the system to recognize and transcribe.
        - **Output Language**: This is the language into which the email content will be translated.

2. **Enter Sender's Mail Details.**
    - Provide the sender's email credentials:

- **Sender's Email ID:** Enter the email address that you wish to use to send the email.
- **Sender's Password**: Enter the password associated with this email account. This information is securely used for authenticating with the mail server to enable sending functionality.

Figure 3.3 Sender details and Mail subject

3. **Speak the Mail Subject.**
    - Click on the “Please Speak the Subject” button to activate the speech recognition feature for the subject field.
        1. Once activated, speak the subject of your email clearly.
        2. The system will process your speech and convert it into text, which will appear in the subject field.
        3. Review the transcribed text to ensure accuracy. You can re-record the subject if needed by clicking the button again.

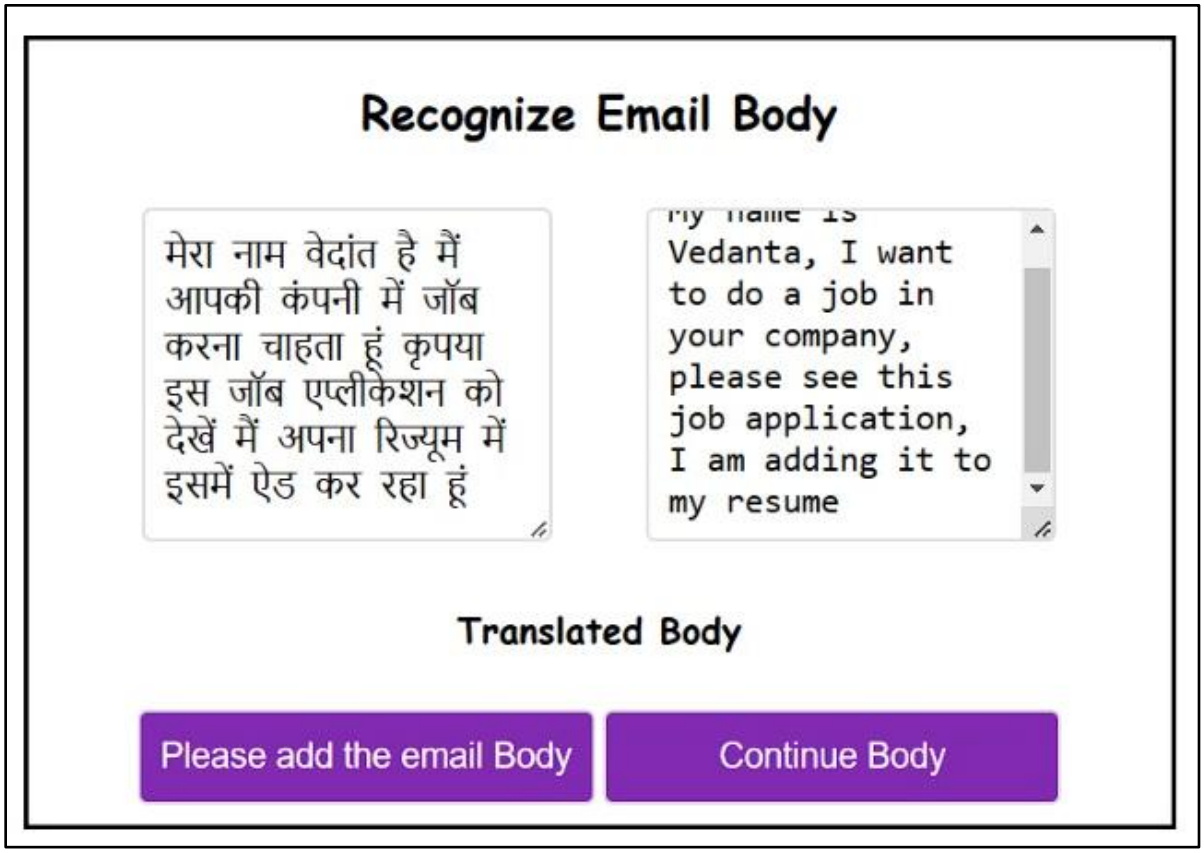

Figure 3.4 Mail body

4. **Compose the Mail Body.**
    - To add content to the body of your email:
        1. Click the “Please Add the Email Body” button to activate speech recognition for the email's body.

2. Speak the content you want to include. The system will transcribe your speech into text and populate the email body field.
3. Review the generated text and make manual corrections if necessary.

- Additionally, you can attach files such as PDFs, images (JPG), or other formats to your email:
    1. Click the "Add Attachments" button to browse and upload files from your system.
    2. Ensure the files are successfully uploaded before proceeding to the next step.

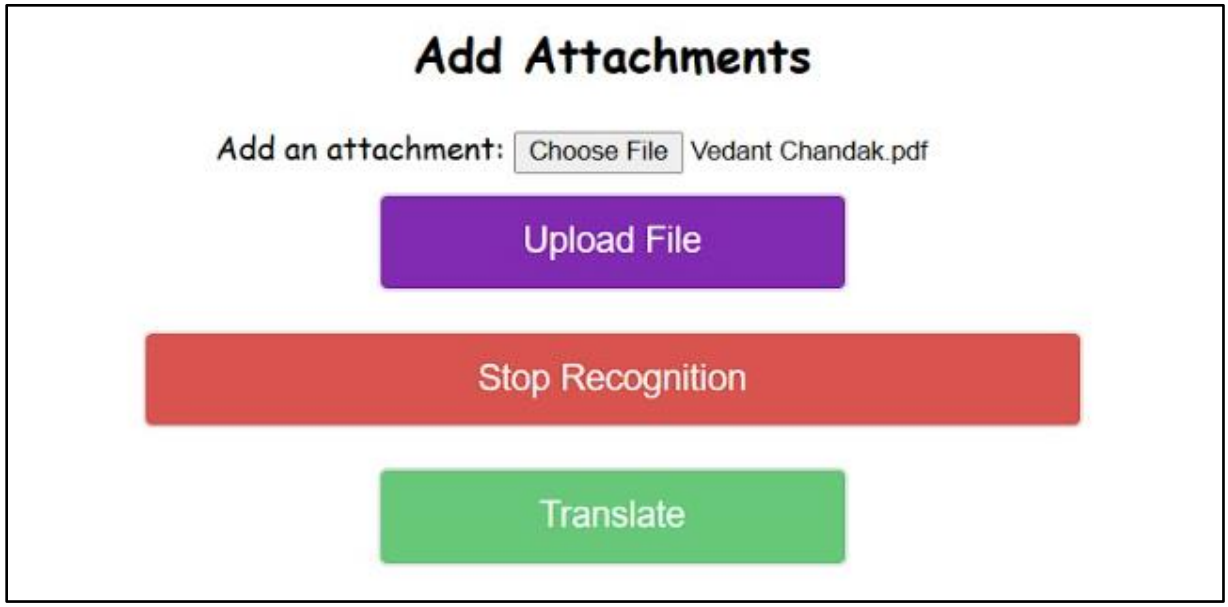

Figure 3.5 Add attachments

5. **Translate the Email**
    - Once you have entered the subject and body:
        1. Click the **"Translate"** button.
        2. The system will translate the subject and body of your email into the output language selected in Step 1.
        3. Verify the translated text to ensure it aligns with your intended message.

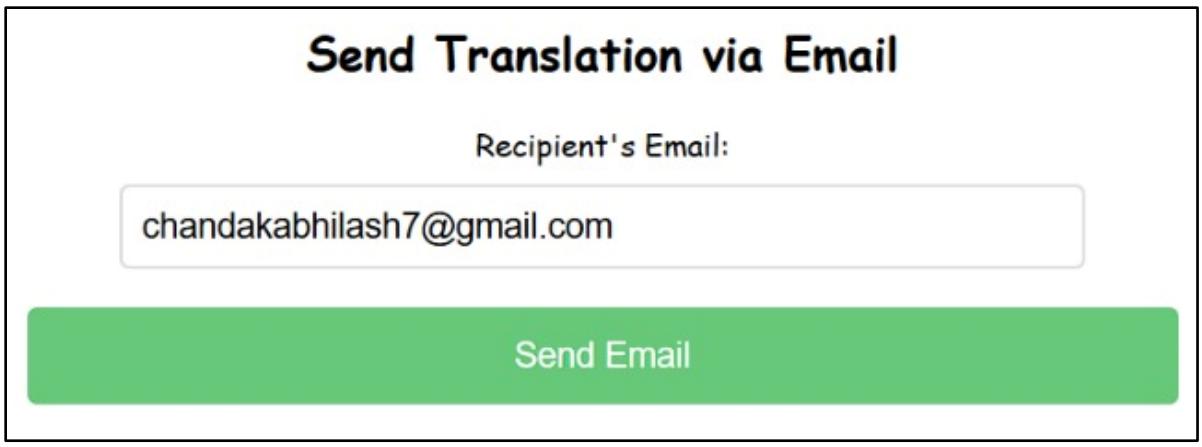

Figure 3.6 Receivers mail id

6. **Add Recipient's Email ID**
    - Enter the email address(es) of the intended recipient(s) in the designated field:
        1. Use commas to separate multiple email IDs if sending to more than one recipient.
        2. Ensure all addresses are valid and properly formatted to avoid delivery issues.

7. **Send the Email.**
    - After completing all the above steps:
        1. Click the "Send Email" button.
        2. WhisperMail will securely send the email, along with any attachments, through the integrated mail service.

WhisperMail offers an efficient, hands-free solution for composing and sending emails using voice commands. By integrating speech-to-text technology with email functionality, it simplifies the process of email creation, translation, and delivery. This tool is not only a time-saver but also enhances accessibility for users who prefer voice-based communication. Whether for personal, professional, or accessibility purposes, WhisperMail streamlines email tasks, making them faster and more convenient for everyone.

3.4 Project Methodology:

The **Prototyping Methodology** involves the development of multiple versions, or prototypes, of a system. Each prototype is built to represent a subset of the system's functionality, allowing the development team to test specific aspects of the software. As each prototype is created, it is evaluated, refined, and enhanced based on insights gained from testing and internal evaluations.

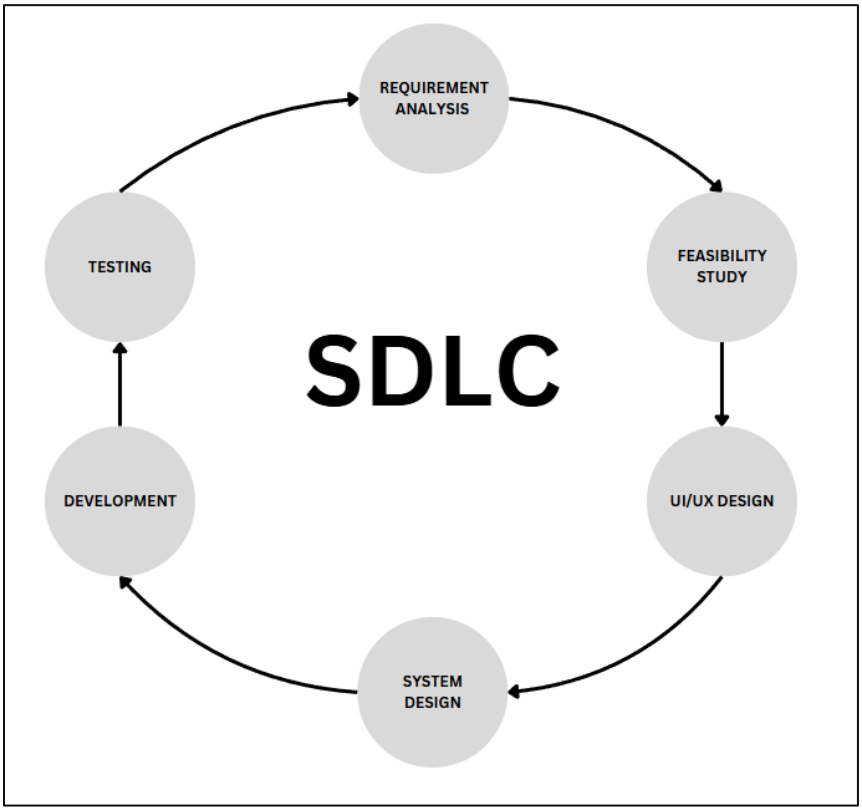

Figure 3.7 SDLC Diagram

1. **Requirement Analysis**
   - **Functional Requirements**:
     - Real-time transcription for composing emails.
     - Email integration to send transcribed content.
     - Basic functionalities for correcting errors, adding punctuation, and formatting the text.

- **Non-Functional Requirements**: You aimed for high speech recognition accuracy, low response times, an intuitive interface, and secure data handling to ensure a reliable and secure application.

2. **Technical Feasibility Study**
   - **Speech Recognition Technology**: Here, you evaluated APIs like Google Speech-to-Text, Gemini. Each option was assessed for factors like accuracy, language support, and speed.
   - **Email API Integration**: You considered APIs such as SMTP, Gmail API, or third-party services (e.g., SendGrid) based on their reliability, security, and scalability.
   - **Technology Stack Selection**:
     - **Frontend (HTML, CSS, JS)**:
       - Contains input fields for user email details, attachment upload, and buttons for sending the email.
       - JavaScript handles events and uses fetch API to send the form data to the Flask backend.
     - **Backend**:
       - Receives the HTTP request with form data and processes it.
       - Validates the data (e.g., checks attachments, sender/recipient emails).
       - Uses an email-sending library (e.g., smtplib or flask-mail) to send the email through an SMTP
     - **Email Server (SMTP Service)**:
       - The backend connects to an SMTP service (like Gmail's SMTP server).
       - Sends the email to the recipient and handles authentication and transmission.
     - **User Interface Updates**:
       - The Flask backend sends a response (success or error) back to the JavaScript.

- JavaScript updates the user interface with an alert showing the status of the operation (email sent or failure).

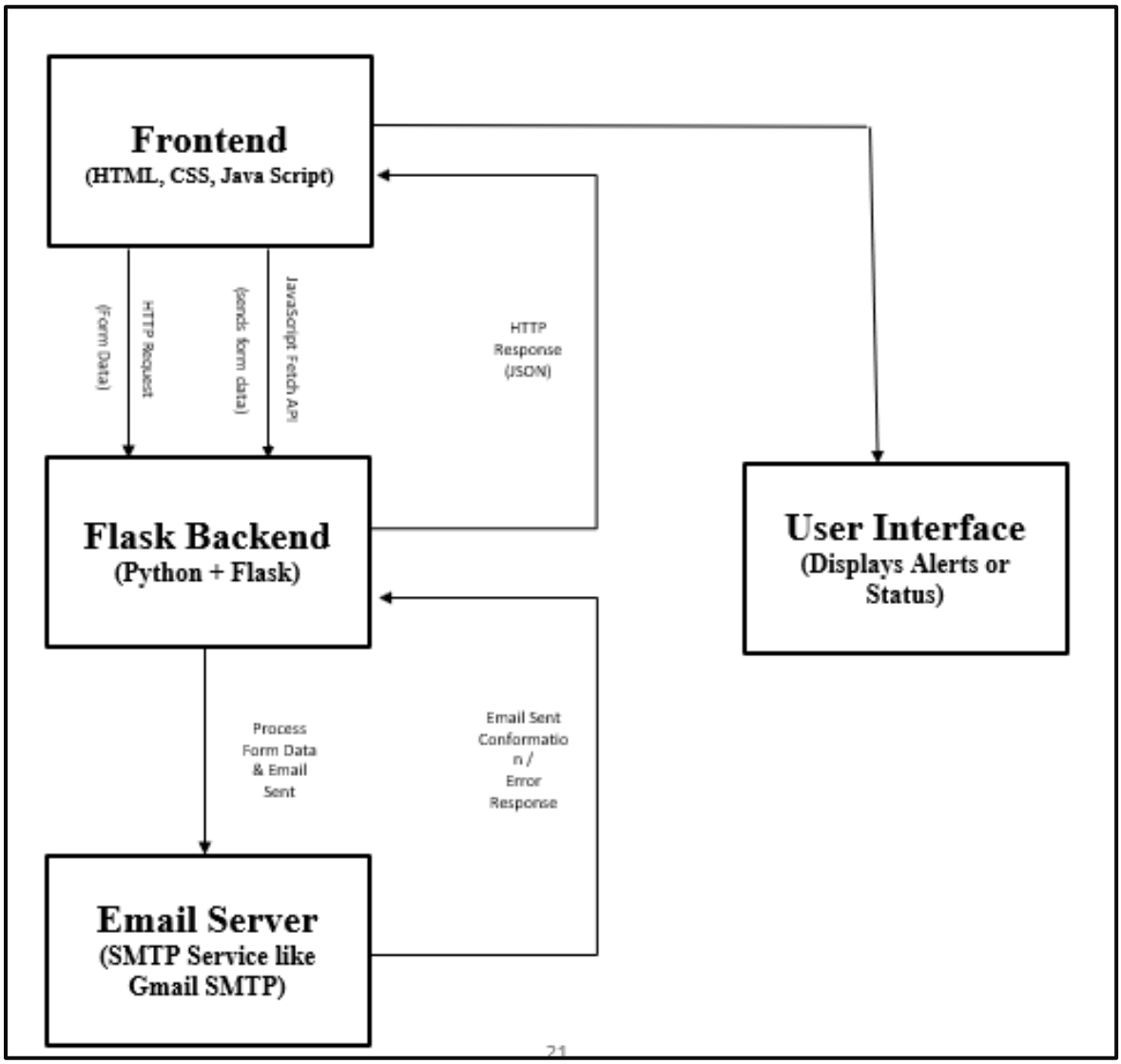

Figure 3.8 Flow Diagram of Tech Stack

3. UI/UX Design

- **Design and Prototyping**: Developed a user-friendly design and prototypes to simulate the user experience.

4. **System Design**
   - **Architecture Design**: The system architecture separated frontend and backend components to ensure scalability and easy maintenance.
   - **Component Design**:
     - **Speech-to-Text Module**: This module was designed to capture and transcribe voice input in real time, populating the email's subject and body fields.
     - **Email Composition and Integration Module**: Enabled users to review and edit transcribed text before sending it through the email API.

5. **Development Phases**
   - **Frontend Development**: Created a responsive interface displaying live transcription, supporting commands for switching fields, and providing controls for starting/stopping voice input.
   - **Backend Development**:
     - **Speech-to-Text Integration**: Integrated the selected speech API, processing and structuring transcribed text for display.
     - **Email API Integration**: Enabled email sending by integrating with the chosen email API.

6. **Testing**
   - **Unit Testing**: Tested individual components like the speech-to-text and email-sending functions.
   - **Integration Testing**: Validated data flow and system functionality end-to-end.
   - **Usability Testing**: Gathered user feedback to refine ease of use, accessibility, and functionality.

speech2.py

```python
import googletrans
import speech_recognition as sr
import gtts
import os

# Define available languages
lang_lib = ["en", "mr", "hi", "ml", "ta", "te"]
lang_dict = {
    "en": "English",
    "mr": "Marathi",
    "hi": "Hindi",
    "ml": "Malayalam",
    "ta": "Tamil",
    "te": "Telugu"
}

def get_language_input(prompt, lang_lib):
    while True:
        lang = input(prompt)
        if lang in lang_lib:
            return lang
        print(f"Invalid selection. Choose from {lang_lib}")

# Select input and output languages
input_lang = get_language_input("Select input language " + str(lang_lib) + ": ", lang_lib)
output_lang = get_language_input("Select output language " + str(lang_lib) + ": ", lang_lib)

# Recognize speech
recognizer = sr.Recognizer()
with sr.Microphone() as source:
```

Figure 3.9 Prototype code 1

speech5.py

```python
import googletrans
import speech_recognition as sr
import gtts
import os
import logging
import tkinter as tk
from tkinter import messagebox

# Setup logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)

# FileHandler for logging to a file
file_handler = logging.FileHandler('translator.log', encoding='utf-8')
file_handler.setLevel(logging.INFO)
file_handler.setFormatter(logging.Formatter('%(asctime)s:%(levelname)s:%(message)s'))

# StreamHandler for console output
console_handler = logging.StreamHandler()
console_handler.setLevel(logging.INFO)
console_handler.setFormatter(logging.Formatter('%(asctime)s:%(levelname)s:%(message)s'))

logger.addHandler(file_handler)
logger.addHandler(console_handler)

# Define available languages
lang_lib = ["en", "mr", "hi", "ml", "ta", "te"]
lang_dict = {
    "en": "English",
    "mr": "Marathi",
```

Figure 3.10 Prototype code 2

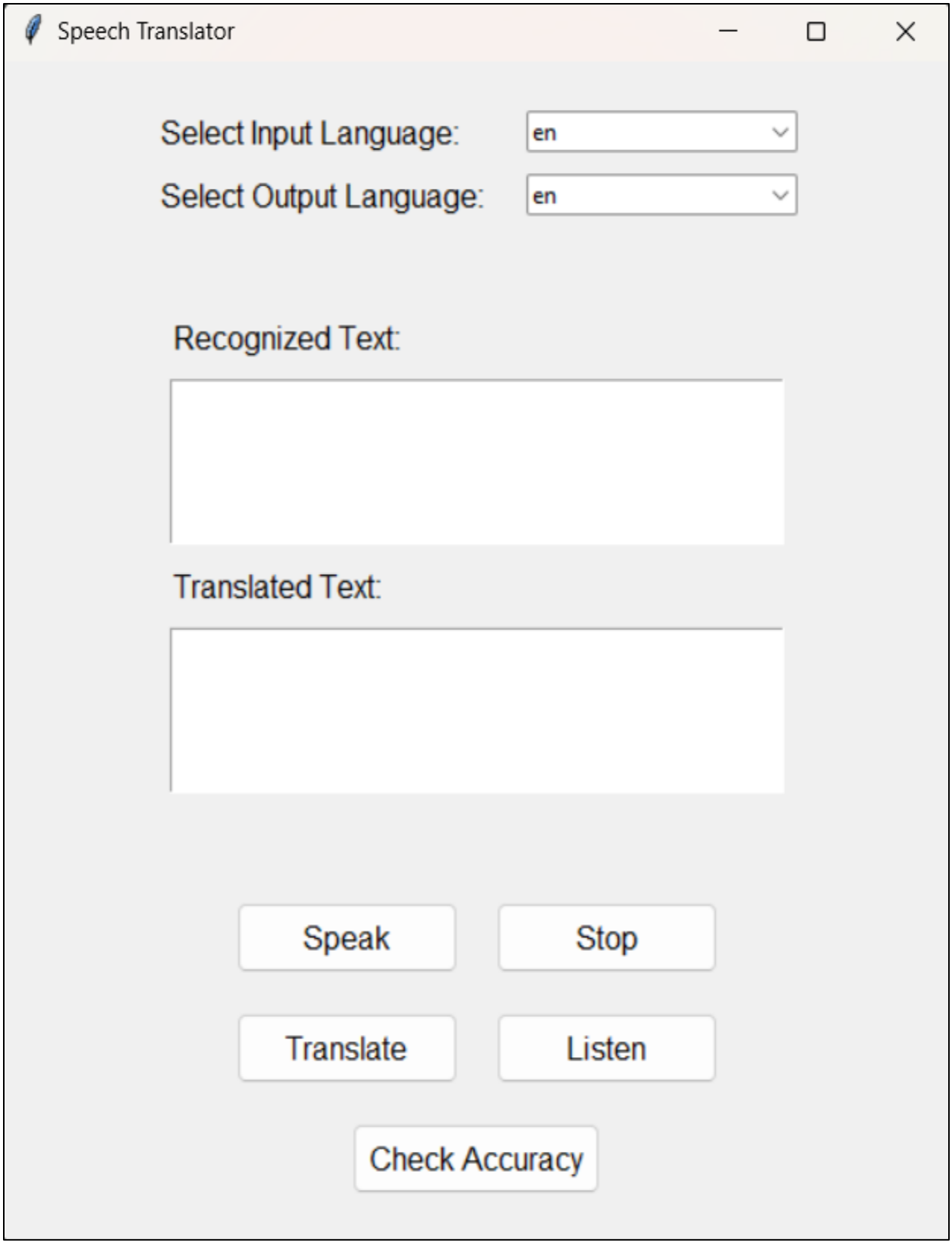

Figure 3.11 Prototype 1

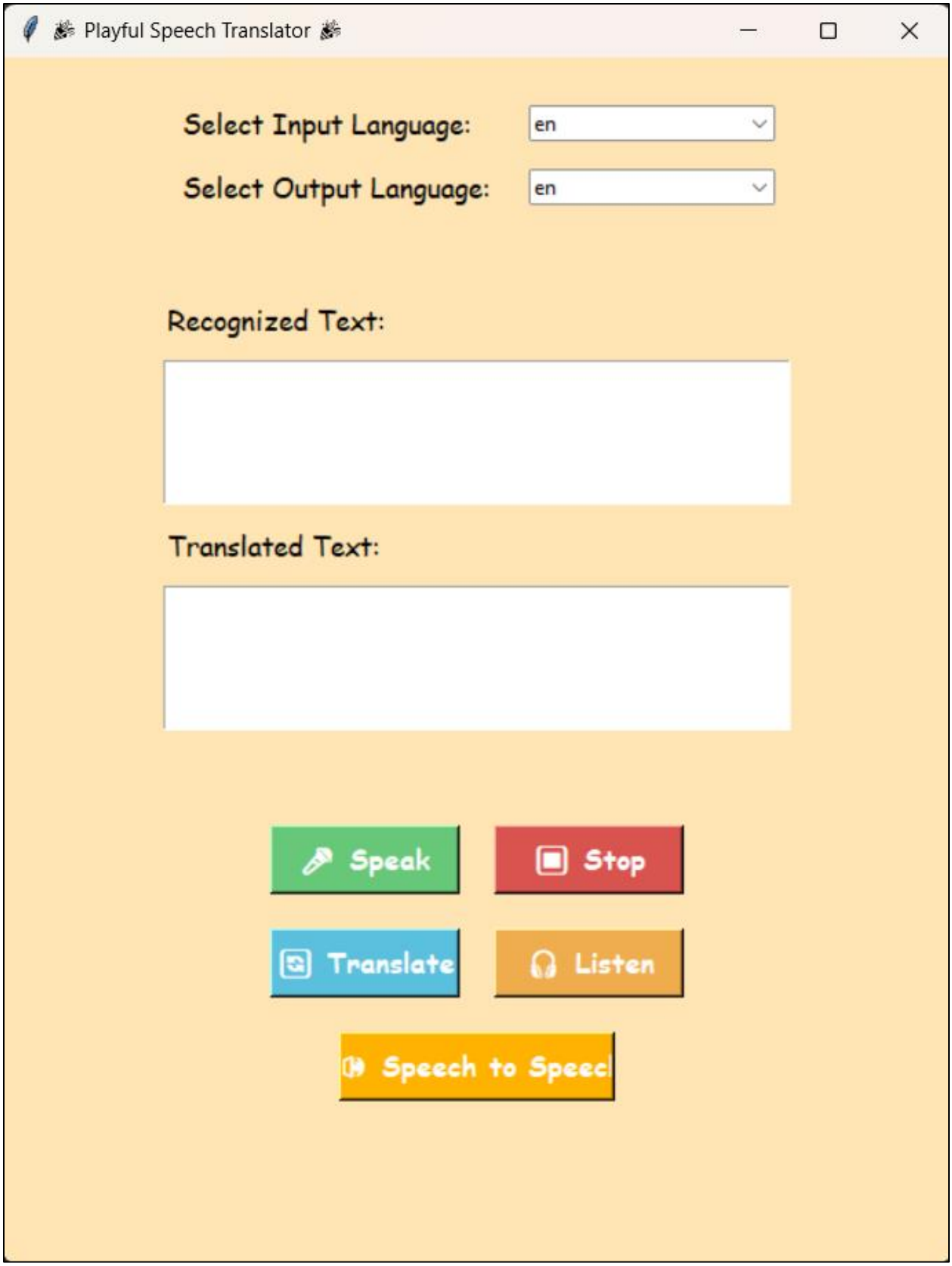

Figure 3.12 Prototype 2

Chapter 4

PERFORMANCE ANALYSIS

❖ **Use Cases:**

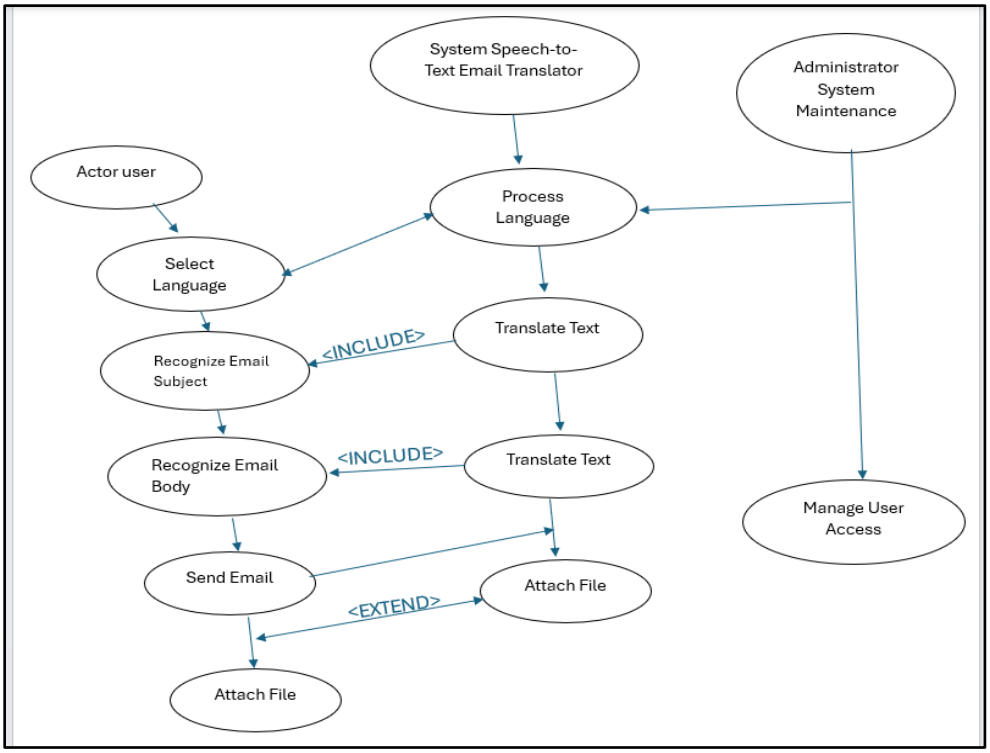

Figure 4.1 Use Case Diagram

1. User (End User):

- A person who interacts with the system to compose emails using speech-to-text functionality.
- Key use cases:
  - Select Language
  - Recognize Email Subject
  - Recognize Email Body
  - Translate Text
  - Send Email
  - Attach File

2. Subsystem:

- The Speech-to-Text Email Translator is the core subsystem handling speech recognition, text translation, and email management.

- Key functionalities:
    - Process Language
    - Translate Text
    - Handle optional file attachments.

3.Administrator:

- A system-level actor responsible for managing and maintaining the subsystem.
- Key responsibilities:
    - Manage User Access.

❖ **Test Cases for Speech-to-Text Email Translator**

Table 4.1 Table of Test Cases

Testcases	**Description**	**Input**	**Expected output**
1	Select a language	Hindi	Language changes to Hindi
2	Recognize email subject content	Speech: "Job Ke Liye arji."	Text: “Job Ke Liye arji.”
3	Empty input	inefficient	inefficient
4	Translate to a valid language	“Hello” Hindi selected	Output:“नमस्ते”
5	Attach a valid file	PDF file: "document.pdf"	File upload successfully
6	UI responsiveness	Test on different devices	UI adjusts properly for mobile, tablet, and desktop
7	Multilingual support	Inputs in Hindi and Marathi	Text correctly translated and displayed

CHAPTER 5

CONCLUSION

WhisperMail is a refined and enhanced version of the original speech-to-text translator, now tailored for real-world email communication. By integrating advanced features like speech recognition, language translation, and email functionality, WhisperMail transforms how users interact with technology. It makes the process of composing and sending emails accessible, and inclusive.

The addition of controls like start/stop buttons, reusable functions ensures that the application is reliable and user-friendly. WhisperMail incorporates email integration to address practical communication needs, making it a robust solution for diverse users.

This project highlights the journey from a foundational idea to a versatile tool that combines voice technology with everyday tasks. Looking ahead, WhisperMail has the potential to grow further.

Reference:

[1] Nicola Bertoldi, Richard Zens, Marcello Federico, and Wade Shen. 2008. Efficient speech translation through confusion network decoding. IEEE Transactions on Audio, Speech, and Language Processing,16(8):1696–1705.

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL:Automatic estimation of 3D human pose and shape from a single image.

[3] Bouck, E., Flanagan, S., Joshi, G., Waseem, S., & Schleppenbach, D. (2011). Speaking math – A voice input, speech output calculator for students with visual impairments. Journal of Special Education Technology, 26(4)

[4] A. Kain and M.W. Macon, "Spectral voice conversion for text-to-speech synthesis", Proceedings of the 1998
IEEE International Conference on Acoustics, Speech and Signal Processing

[5] Alshawi, Hayan, David Carter, Steve Pulman, Manny Rayner, and Björn Gambäck. 1992. "English-Swedish Translation Dialogue Software." In Translating and the Computer, 14, pages 10-11. Aslib: London, November, 1992.

[6] Clarifai (FullStackAIPlatform)

[7] NVIDIA (GitHub)

[8] https://youtu.be/LEDpgye3bf4?si=7DiFJz_VeGDLDNKG

[9] https://youtu.be/SFGIKucaOZA?si=Dra47v14Mnv9A0nU

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all those who have supported and guided us throughout the completion of this project.First and foremost, we extend our sincere thanks to Dr. Seema R. Chaudhary, our project guide, for her invaluable guidance, continuous support, and encouragement throughout this project. We are also deeply grateful to the teaching and non-teaching staff, as well as the library staff, for their assistance and support. Their contributions and resources have been instrumental in facilitating the research and development of this project. We would like to convey our special thanks to Dr. S. L. Kasar, Head of the Department, for her support and for providing the necessary resources and environment to pursue this project. Furthermore, we are immensely thankful to Dr. N. G. Patil, Director of the Maharashtra Institute of Technology, Chh. Sambhajinagar (Aurangabad).

Thank you all for your unwavering support and guidance.

Vedant Chandak (Roll No: BT4104)

Tushar Katore (Roll No: BT4115)

Vaibhav Rathod (Roll No: BT4126)