Email Interface For Blind People

1Sejal Gurkhe, 2Ishita More
1Student, 2Student
1Sardar Patel Institute of Technology,
2Sardar Patel Institute of Technology

Abstract - **The usage of the current interface of e-mail is quite easy and lucid for regular users. However, the current system is not operable by users with visual defects as the avail-able systems are based on the visual perceptions. Therefore, the proposed project focuses on enhancing/extending the current system with the functionalities like com-posing, reading, sending and receiving emails through voice based interaction made by users with visual defects. Users facing language problems may also use the app because the app offers illiterate people the possibility of having a conversation in their mother tongue (marathi).**

keywords **- Speech recognition, Text to Speech, Speech to text, Email, Translator**

I.INTRODUCTION

There are almost certainly more than one billion email accounts. There are over 2.586 billion email subscribers worldwide, including both business and consumer users. One of the major annoyances that develops is the requirement for visual ability to access emails or any webpage on the internet. This makes technology useless because a person with visual impairments cannot access the services offered by the internet.

The introduction of technologies like screen readers, automatic voice recognition, speech-to-text, and text-to-speech made things easier for the blind, though not entirely. It is not necessary for the individual using this project to recall every important keyboard location. Voice command interactions will underpin every action. Additionally, since the system will accept and take into account commands given through voice alone, the blind person need not worry about which mouse clicks to make in order to use a certain service.

This project proposes creating a desktop application in Python that is built specifically for individuals who are blind or visually impaired. They can read and send emails on their own using a voice-based emailing service that is managed by this application. Here, users can speak certain keywords to carry out specific tasks, such as Read, Send, Compose Mail, etc. Additionally, the system enables user connection on social media platforms. The application offers illiterate people the choice to communicate in their local language, so users who have language barriers can also use it (marathi).

A person who seems to be blind can actively account for mail services via the voice mail system. Because third-party applications are unsafe, visually impaired users do not need any support from them in order to access mail. Every user of this application will be able to manage their own accounts entirely using voice inputs through speech to text and text to speech modules. The user will always be required to perform the steps necessary by this system based on the related commands. The motivation for this project's development is the fact that many operations will be voice-based entirely, with little to no keyboard or mouse use.

II. OBJECTIVES

1. To create an e-mail system that will even allow a person with a visual impairment and no previous experience to use the services to communicate.
2. There is no requirement to use a keyboard.
3. It will only work with speech-to-text conversion.
4. Assist illiterate users in using their own language for communication.

III. PROBLEM STATEMEBT

A web-based application created in Python for the Voice Mail System project facilitates access to email for people who are visually impaired. The suggested system aims at providing voice command interaction as well as e-mail related activities such as writing, reading, sending and receiving. The system will be able to receive orders from the user, which they will then complete as specified. With social media services such as WhatsApp and Telegram, users can send written emails directly to others. Users who have trouble communicating in English can use the mail interface in their mother tongue (here, we tried the marathi language).

IV. RELATED WORK

The mail services that are accessible today are of no use to the people who are visually impaired. This is because these systems are difficult to operate in any way as it cannot provide any audio feedback to readout the contents for them. As they are insufficient to imagine things that are present on the screen, they find it difficult to perform operations such as performing mouse clicks specifically. Although there are screen readers available, they impose some or the other kind of difficulty on them. In short, screen readers read out the text on the screen for them, and they must use a keyboard to respond to it. Therefore, the user must be aware of the key locations on the keyboard in order to achieve this. An individual who has never used a computer will therefore never be able to use such a system.

V. Literature Survey

The paper [1] suggests a web application that completely eliminates the idea of keyboard shortcuts and also suggests the use of screen readers, which may help reduce the stress associated with storing keys on the keyboard. Only instructions supplied by Google's voice recognition technology should be followed by the user. Further, the user may be required to provide the information orally where indicated. Furthermore,paper [2] focused Receiving unseen emails with vocal output of the sender's mail ID, subject, and message. Also, implement a chatbot as a result of effective user and system interaction.

However the paper [3] proposes learning and using the keyboard shortcuts and proposed using voice authentication rather than the customary username and password. The paper [4], presents the VoiceMail system architecture with an accessibility gap for the blind population to key electronic communication channels like Email. and assess the architecture by contrasting proposed GUI's performance with that of the current Gmail GUI. IVR[5]: It's a voice-activated response system. It provides the user with voice interaction with the system. It analyzes and synthesizes the user's voice and responds through voice, text, e-mail, etc. project's [6] goal is to create a voice-activated system for visually impaired people that will transmit information as audio and allow the user to: 1. Send and Receive emails 2. Use the news of the day 3. A weather report 4. Keep a personal blog or online journal.

Aditionally, a modular approach to enhancing web-based accessibility for the blind to maximize usability and ensure a hassle-free experience for the user, the virtual assistant is operating system independent and does not rely on keyboard inputs from the user is suggested[7] . The three currently in use modules' implementation methods and system design are shown. For speedy and accurate user query responses, the Wikipedia module uses a BERT model on the SQuAD dataset.

Digital Assistant for the Visually Impaired[8], or DAVID, is an android-based application that uses Google's speech recognition and speech synthesis capabilities to create a digital assistant that can communicate with the blind and is capable of detecting, recognising Latin characters or text from real- world objects and providing a real-time audio feedback of the detected text to the blind user.

However the paper[9] suggested creating a voice-based mailing service so that individuals can read and send letters alone, with no assistance. Here, users must enter certain keywords that will carry out specific tasks, such as Read, Send, Compose Mail, Address Book, etc. A blind person can quickly and effectively access emails using this EMAIL system. With the suggested method in paper [10] blind users can access their mail on Android devices without the aid of a third party. They require the help of other people when installing the application and registering users, which are the first steps.

VI. System Design

When the system starts, it will redirect to the dashboard. Where the modules will be displayed, using text to speak system will read all the functionality. Then the user will speak out which function should be performed. After that, using the speak to text module, the system will get to know the function to perform. It will validate the number and proceed for respective tasks. (Fig. 1 & 2)

Functions

1. *Send Mail*

   In the Compose Mail module, the system requests the sender's email ID, subject line and body. After every step, a TTS module then called Senders Email Id, Subject and Body mail to be read out loud by the system and to be confirmed by the user much the same. The system then transmit the text to the email module and the email is appropriately formatted. After Composing Mail, System will ask the user whether to forward the mail on whatsapp or/and telegram. (Fig. 2)

2. *Get Mail Status*

   In this function, the total number of mails in the mailbox is shown. (Fig. 1)

3. *Search Mail*

   In Search Mail, using the subject the user can search mail. Users have to speak about the subject. The System will search. If found, using the text to speech module system will read the mail. Otherwise, System will directly say mail is not found. (Fig. 2)

4. *Latest Mail*

   In Latest Mail, users have to specify from which folder he/she wants to get the latest mail. After the command the latest mail will be read using text to speech. (Fig. 1)

5. *Sending Mail through Marathi Language*

In the Compose Mail Module, System will ask for the sender's email ID, subject line and body of the mail. User will speak in Marathi and the system will translate it in English. After every step, a TTS module then called Senders Email Id, Subject and Body mail to be read out loud by the system and to be confirmed by the user much the same. The system then transmit the text to the email module and the email is appropriately formatted. (Fig. 3)

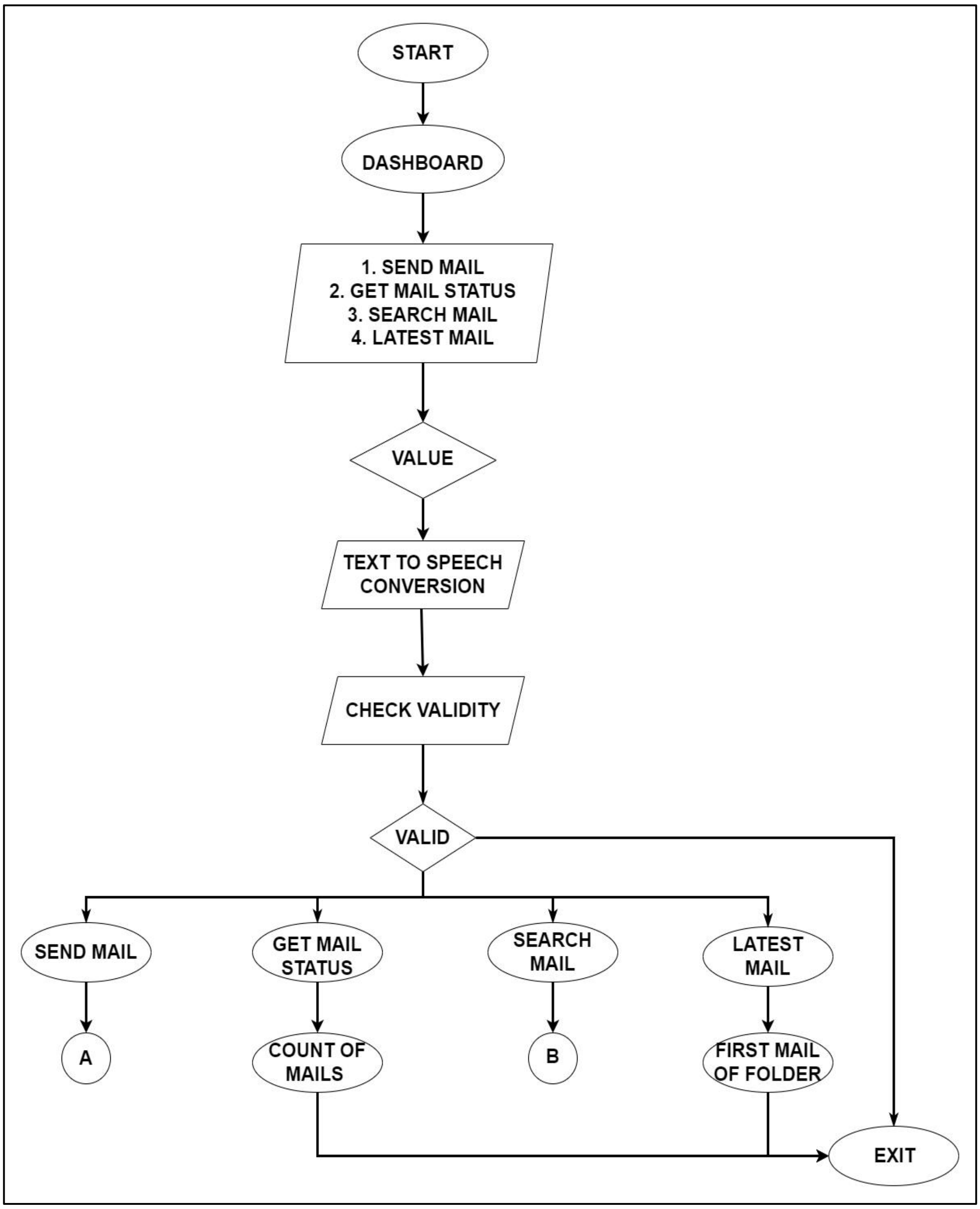


Figure 1. System Diagram

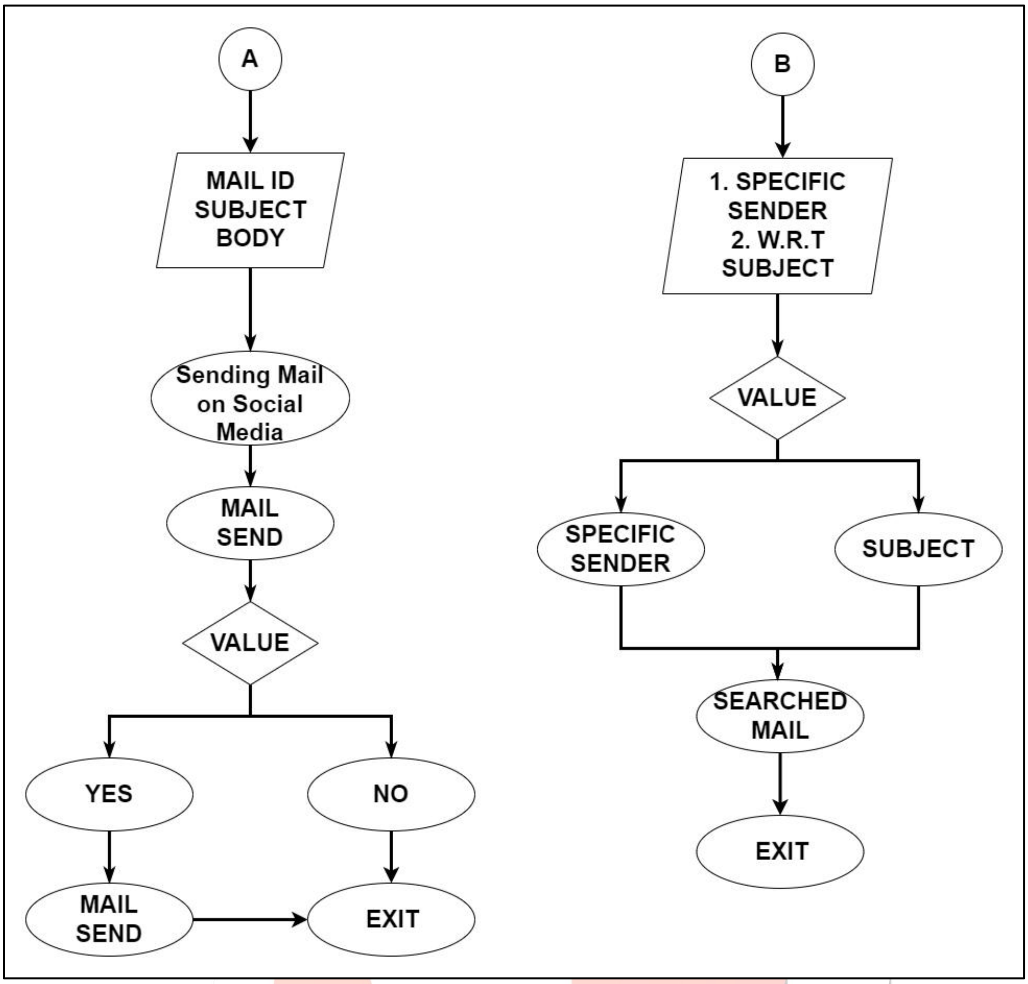


Figure 2. System Diagram

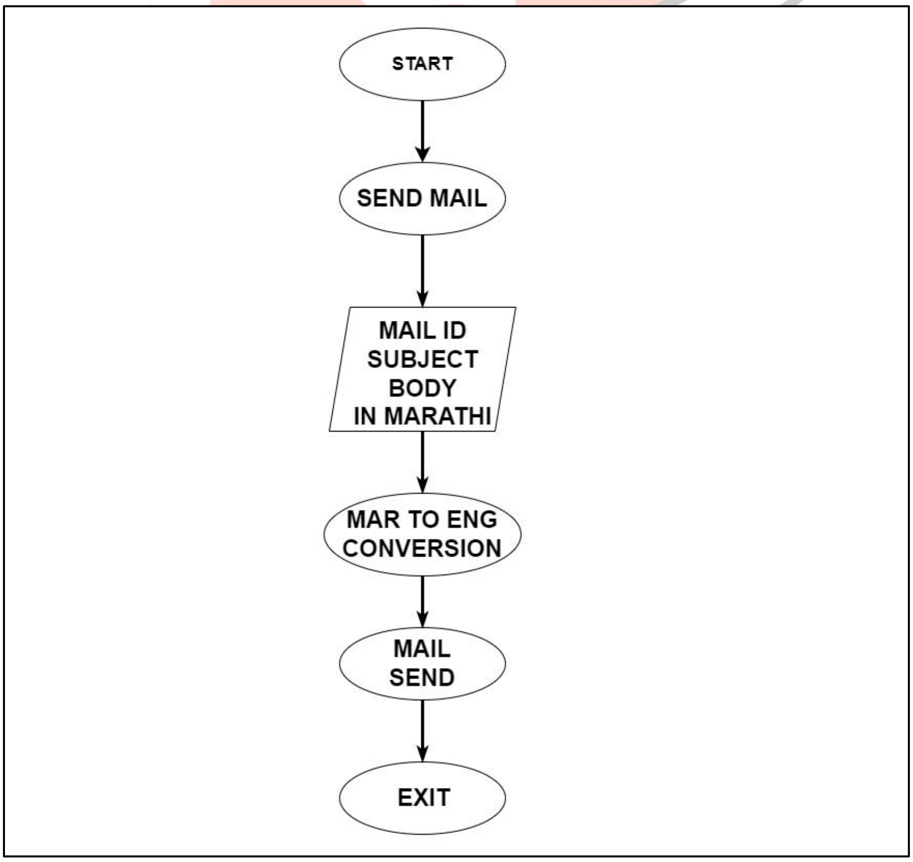


Figure 3. System Diagram

Design and Methodology

The architecture of this project consists of three parts: Hardware Configuration, Software Configuration and Operation. Let's walk through each section in detail. For the implementation of this project, we will require hardware products such as a desktop to run the application and a microphone to communicate with the application. The program is under development using coding languages like Python and will be using Flask. As well, Google Speech API will be used for communication. In addition, speech to text conversion will be performed while the user inputs the data.

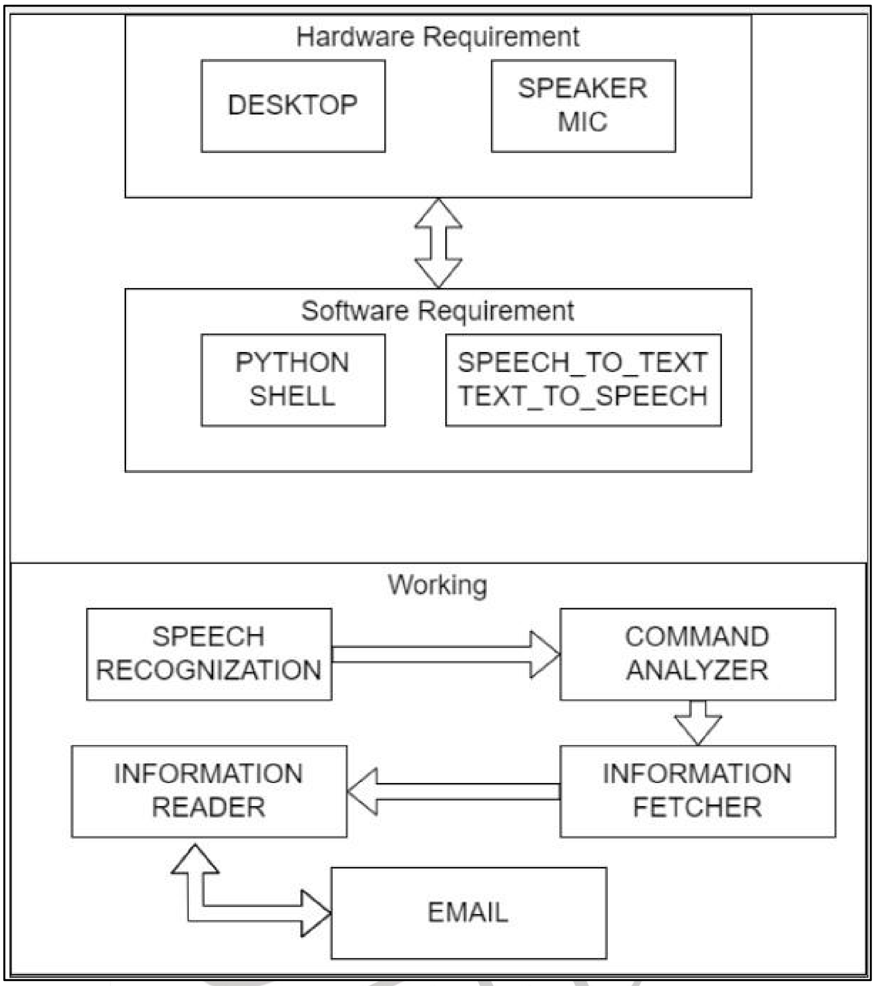


Figure 4. Design & Methodology

Tech Stack

1. Language

   Python

   A dynamically semantic, object-oriented, high-level programming language. Python is well-known for its high-level, built-in data structures, dynamic typing, and dynamic binding. It is also used frequently for rapid application development and as a scripting or glue language to connect existing components. Python’s easy-to-learn syntax and focus on readability reduce the expenses associated with maintaining programs. Additionally, Python’s support for modules and packages streamlines the process of writing modular programs and reusing code.

2. API

   The Application uses following APIs:

   1. *Playsound API:*

      The playsound module contains only a single function named playsound(). It requires one argument: the path to the file with the sound we have to play. It can be a local file, or a URL. There's an optional second argument, block, which is set to True by default.

   2. *Speech Recognition:*

      It is used for individuals to perform actions through voice instead of using a keyboard. This software takes input through voice and saves in mp3 format. After saving, it performs a particular task and deletes after the function is over. The user dictates the messages and the system accepts the message in voice format.

3. *IMAP:*
   Internet message access protocol is used to fetch mails from Gmail so that the receiver can read the messages easily. The received messages are in the gmail itself IMAP helps the user to connect to gmail.

4. *SMTP:*
   Simple mail transfer protocol helps to connect to Gmail. It helps in composing mails. It also acts as a connection to gmail which helps in sending mails.

5. *Email:*
   An email API—short for email Application Programmable Interface—is an interface that allows developers to connect an application or service to an email service provider and use the provider's functionality (including sending email, creating lists, or pulling email stats) without having to build it themselves.

6. *Indic transliteration:*
   Transliteration tools to convert text in one indic script encoding to another.

7. *Pywhatkit:*
   Python offers numerous inbuilt libraries to ease our work. Among them pywhatkit is a Python library for sending WhatsApp messages at a certain time, it has several other features too. Some features of pywhatkit module are Send WhatsApp messages, Play a YouTube video, Perform a Google Search,etc.

8. *Telegram Bot*:
   A Telegram bot is automated software created to help a user meet their needs with minimal interaction with the system and deliver the best results by understanding what the user says.

9. *Googletrans:*
   Google Translate (Googletrans) is a multilingual neural machine translation service developed by Google to translate text, documents and websites from one language into another.

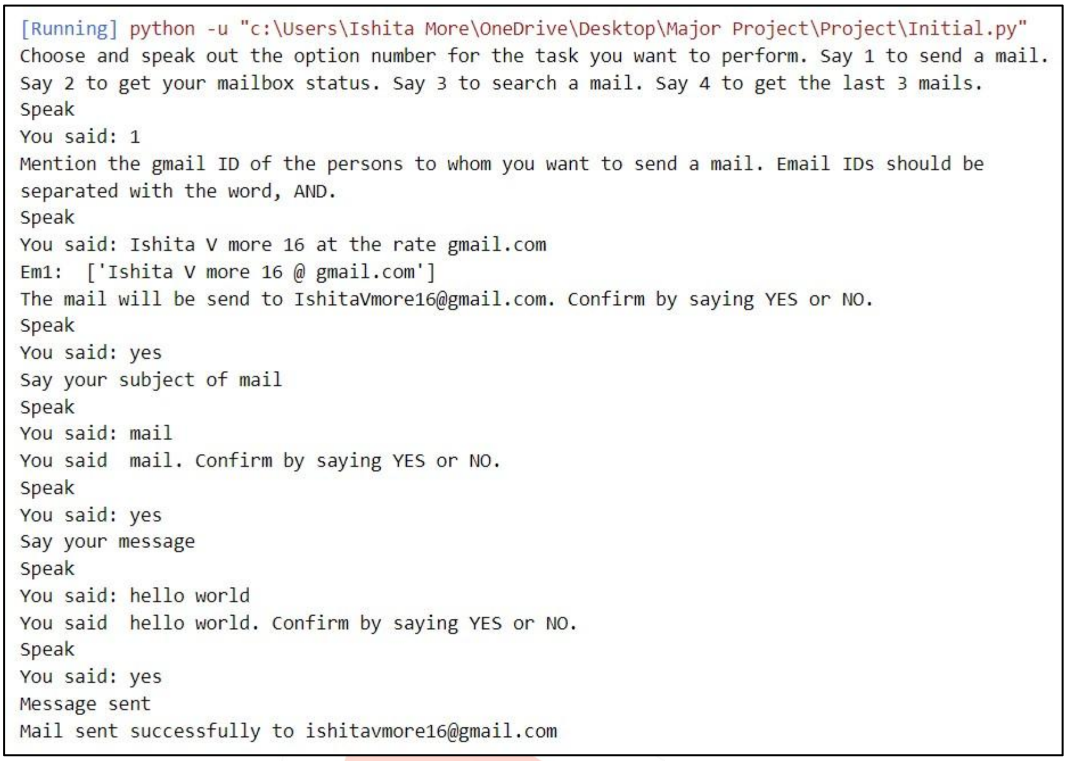
3. Framework

   Flask
   Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

VII. Implementation & Result

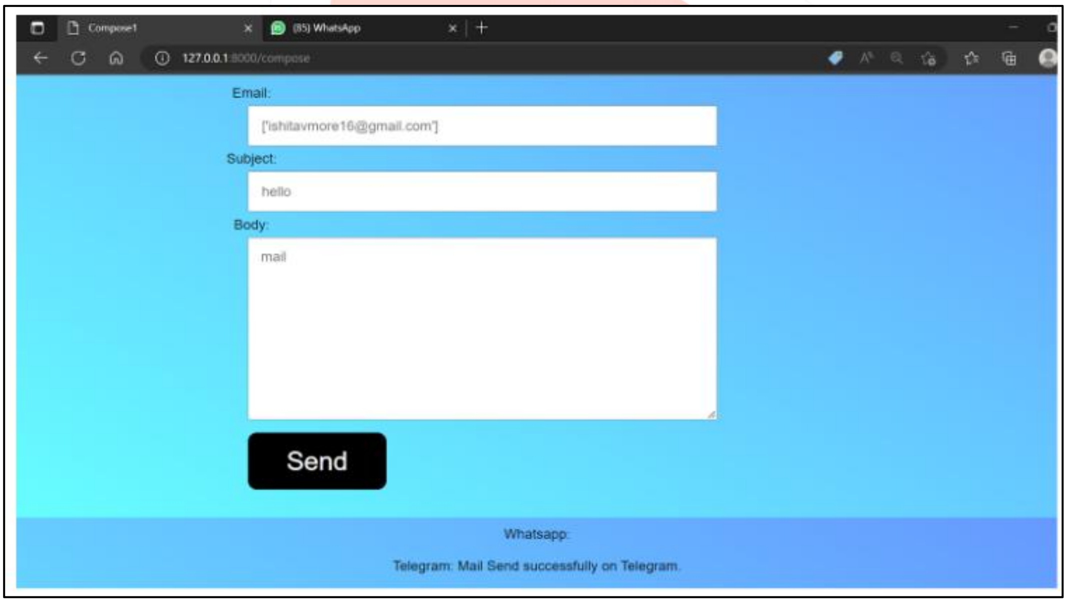
1. *Compose Mail*

   English
   In the Compose Mail module, the system requests the sender's email ID, subject line and body. After every step, a TTS module then called Senders Email Id, Subject and Body mail to be read out loud by the system and to be confirmed by the user much the same. The system then transmit the text to the email module and the email is appropriately formatted. (Fig.5 & 6)



```
[Running] python -u "c:\Users\Ishita More\OneDrive\Desktop\Major Project\Project\Initial.py"
Choose and speak out the option number for the task you want to perform. Say 1 to send a mail.
Say 2 to get your mailbox status. Say 3 to search a mail. Say 4 to get the last 3 mails.
Speak
You said: 1
Mention the gmail ID of the persons to whom you want to send a mail. Email IDs should be
separated with the word, AND.
Speak
You said: Ishita V more 16 at the rate gmail.com
Em1:  ['Ishita V more 16 @ gmail.com']
The mail will be send to IshitaVmore16@gmail.com. Confirm by saying YES or NO.
Speak
You said: yes
Say your subject of mail
Speak
You said: mail
You said  mail. Confirm by saying YES or NO.
Speak
You said: yes
Say your message
Speak
You said: hello world
You said  hello world. Confirm by saying YES or NO.
Speak
You said: yes
Message sent
Mail sent successfully to ishitavmore16@gmail.com
```

Figure 5. Compose Mail (English)



Email: ['ishitavmore16@gmail.com']

Subject: hello

Body: mail

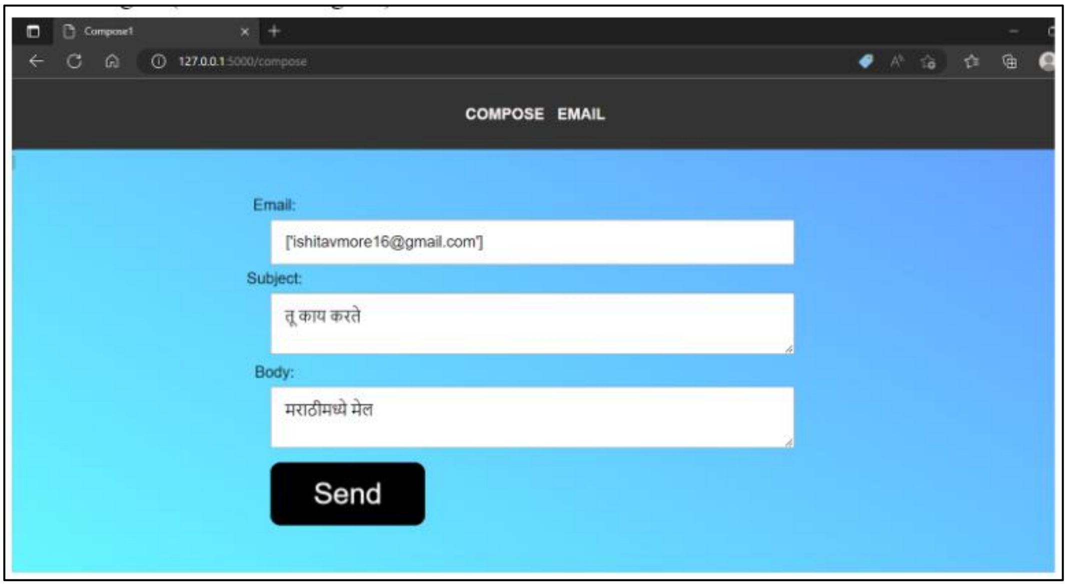
Whatsapp:

Telegram: Mail Send successfully on Telegram.

Figure 6. Compose Mail (English)

Marathi

In Compose Mail Module, System will ask for sender's email ID, subject line and body of the mail. User will speak in Marathi and the system will translate it in English. After every step, a TTS module then called Senders Email Id, Subject and Body mail to be read out loud by the system and to be confirmed by the user much the same. The system then transmit the text to the email module and the email is appropriately formatted. (Fig. 7)



COMPOSE EMAIL

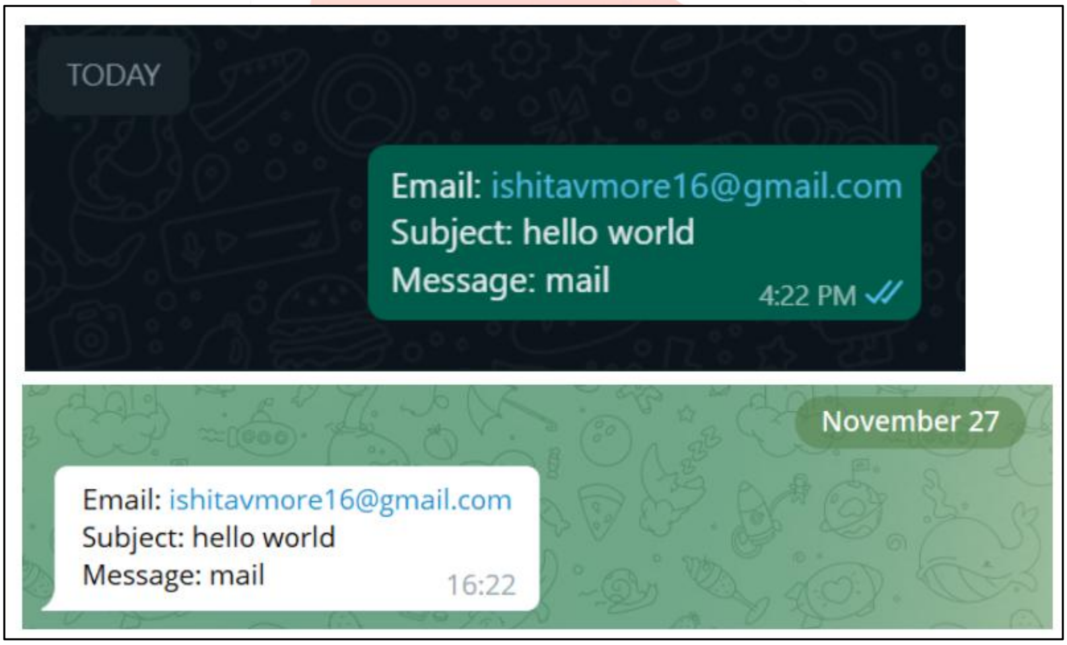
Email: ['ishitavmore16@gmail.com']

Subject: तू काय करते

Body: मराठीमध्ये मेल

Figure 7. Compose Mail (Marathi)

2. *Sending Mail on Social Media(Whatsapp, Telegram)*
   After Composing Mail, System will ask the user whether to forward the mail on whatsapp or/and telegram. This is done using Pywhatkit and Telegram Bot respectively. (Fig. 8). To forward mail on whatsapp the user must be logged in on whatsapp web.

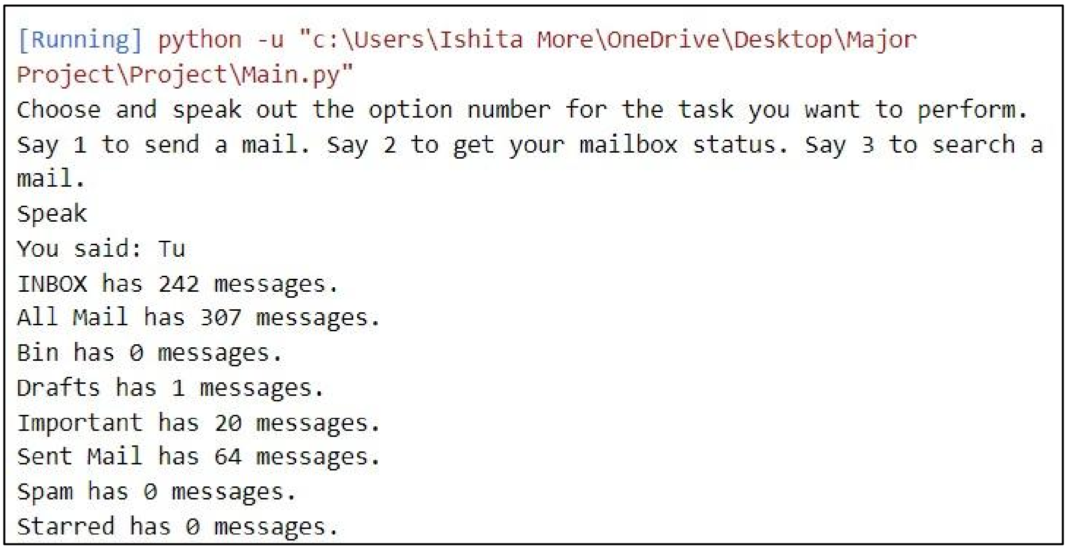


Email: ishitavmore16@gmail.com
Subject: hello world
Message: mail
4:22 PM

Email: ishitavmore16@gmail.com
Subject: hello world
Message: mail
16:22

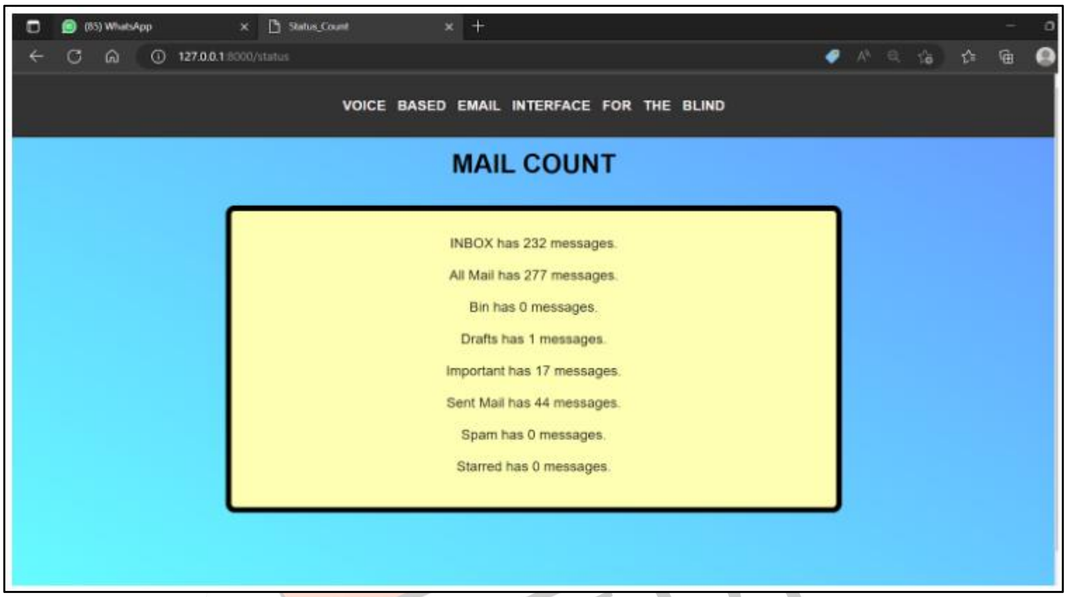
Figure 8. Mail Sent on Social Media

3. *Get Mail Status*
   Here, the system will provide the total count of mails in all the folder in the form of TTS. (Fig. 9 & 10)



```
[Running] python -u "c:\Users\Ishita More\OneDrive\Desktop\Major
Project\Project\Main.py"
Choose and speak out the option number for the task you want to perform.
Say 1 to send a mail. Say 2 to get your mailbox status. Say 3 to search a
mail.
Speak
You said: Tu
INBOX has 242 messages.
All Mail has 307 messages.
Bin has 0 messages.
Drafts has 1 messages.
Important has 20 messages.
Sent Mail has 64 messages.
Spam has 0 messages.
Starred has 0 messages.
```

Figure 9. Get Mail Status



MAIL COUNT

INBOX has 232 messages.

All Mail has 277 messages.

Bin has 0 messages.

Drafts has 1 messages.

Important has 17 messages.

Sent Mail has 44 messages.

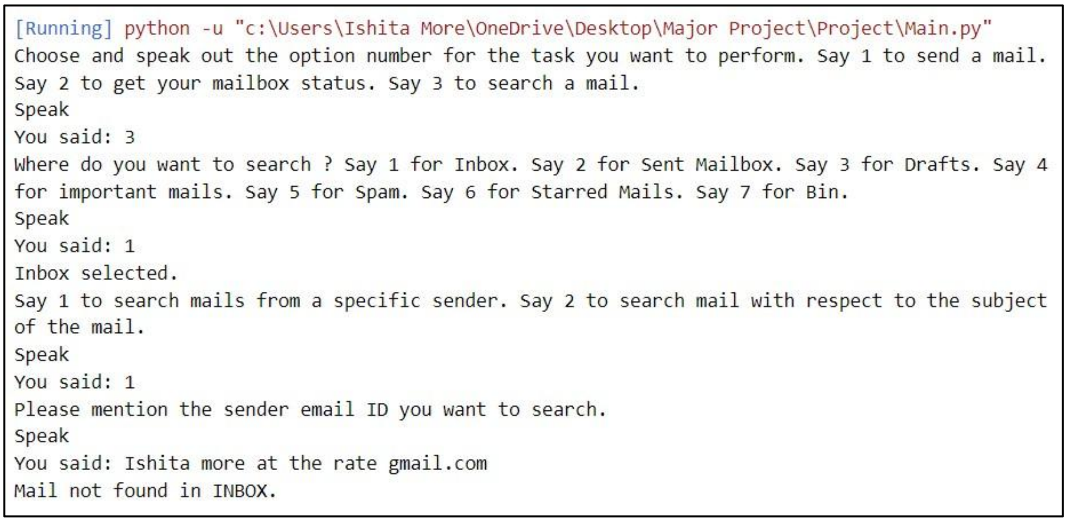
Spam has 0 messages.

Starred has 0 messages.

Figure 10. Get Mail Status

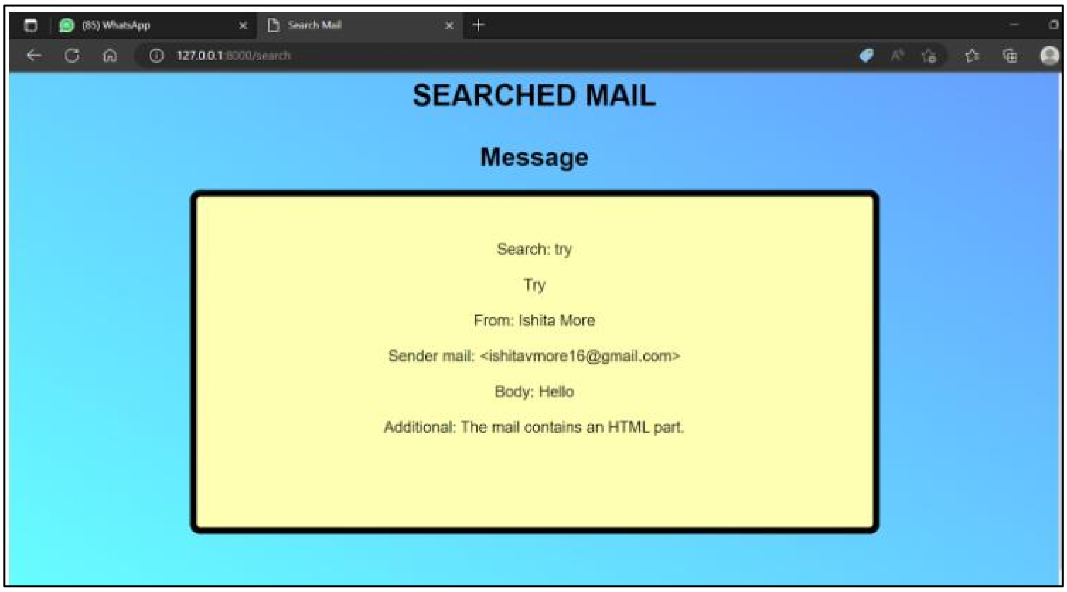
4. *Search Mail*

In Search Mail, using the subject the user can search mail. Users have to speak about the subject. The System will search. If found, using the text to speech module system will read the mail. Otherwise, System will directly say mail is not found. (Fig. 11 & 12)



```
[Running] python -u "c:\Users\Ishita More\OneDrive\Desktop\Major Project\Project\Main.py"
Choose and speak out the option number for the task you want to perform. Say 1 to send a mail.
Say 2 to get your mailbox status. Say 3 to search a mail.
Speak
You said: 3
Where do you want to search ? Say 1 for Inbox. Say 2 for Sent Mailbox. Say 3 for Drafts. Say 4
for important mails. Say 5 for Spam. Say 6 for Starred Mails. Say 7 for Bin.
Speak
You said: 1
Inbox selected.
Say 1 to search mails from a specific sender. Say 2 to search mail with respect to the subject
of the mail.
Speak
You said: 1
Please mention the sender email ID you want to search.
Speak
You said: Ishita more at the rate gmail.com
Mail not found in INBOX.
```

Figure 11. Search Mail



SEARCHED MAIL

Message

Search: try

Try

From: Ishita More

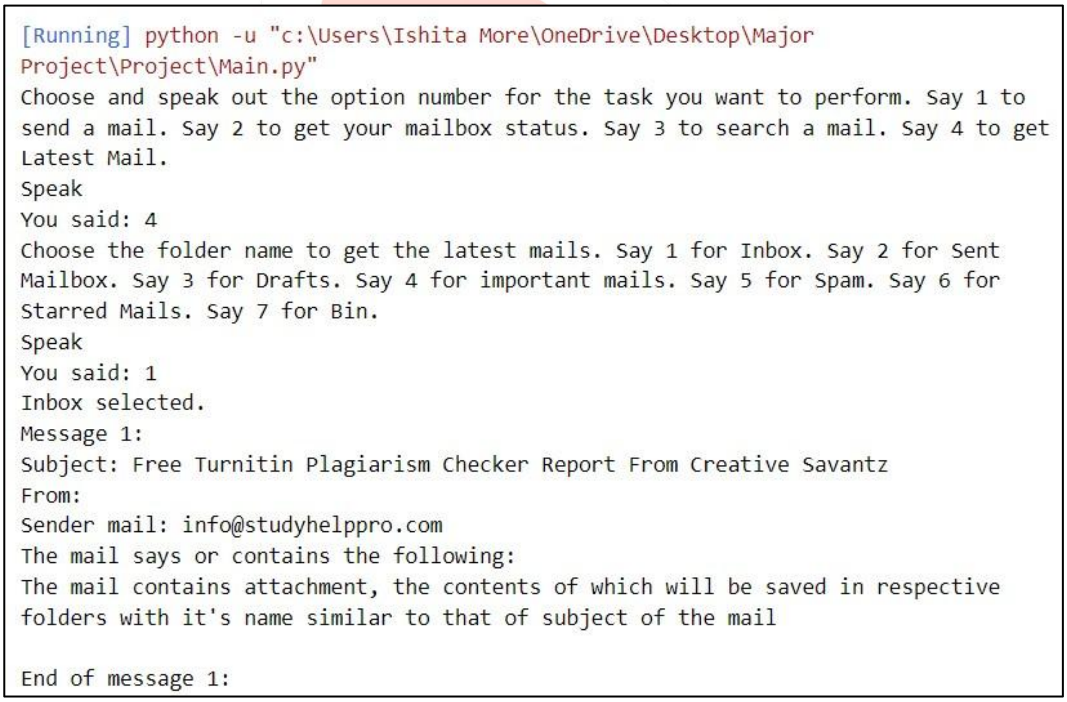
Sender mail: <ishitavmore16@gmail.com>

Body: Hello

Additional: The mail contains an HTML part.

Figure 12. Search Mail

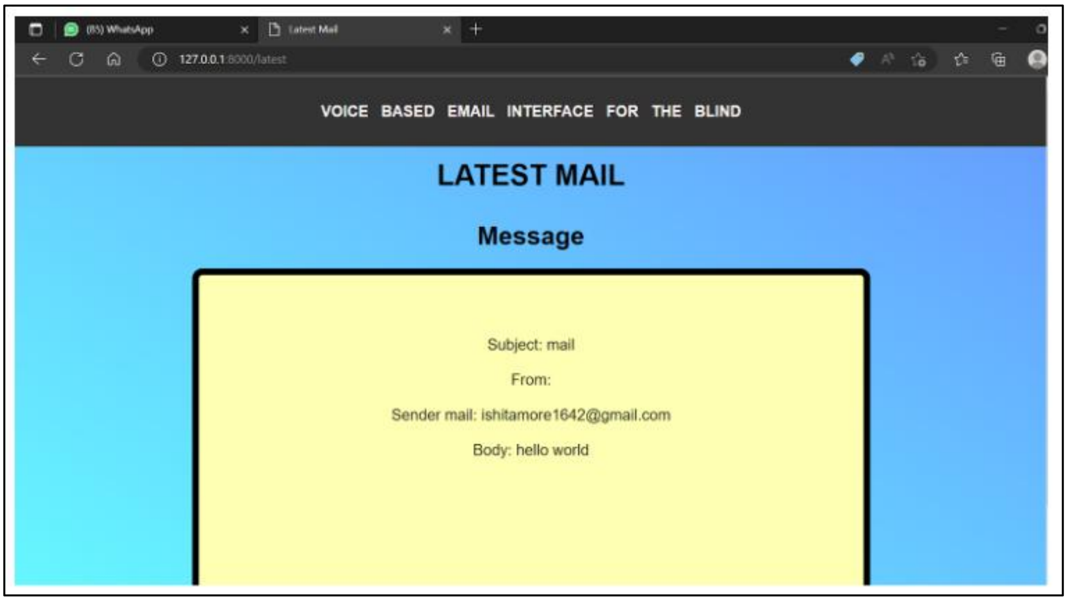
5. *Latest Mail*
   In Latest Mail, users have to specify from which folder he/she wants to get the latest mail. After the command the latest mail will be read using text to speech. (Fig. 13 & 14)



```
[Running] python -u "c:\Users\Ishita More\OneDrive\Desktop\Major
Project\Project\Main.py"
Choose and speak out the option number for the task you want to perform. Say 1 to
send a mail. Say 2 to get your mailbox status. Say 3 to search a mail. Say 4 to get
Latest Mail.
Speak
You said: 4
Choose the folder name to get the latest mails. Say 1 for Inbox. Say 2 for Sent
Mailbox. Say 3 for Drafts. Say 4 for important mails. Say 5 for Spam. Say 6 for
Starred Mails. Say 7 for Bin.
Speak
You said: 1
Inbox selected.
Message 1:
Subject: Free Turnitin Plagiarism Checker Report From Creative Savantz
From:
Sender mail: info@studyhelppro.com
The mail says or contains the following:
The mail contains attachment, the contents of which will be saved in respective
folders with it's name similar to that of subject of the mail

End of message 1:
```

Figure 13. Latest Mail



VOICE BASED EMAIL INTERFACE FOR THE BLIND

LATEST MAIL

Message

Subject: mail

From:

Sender mail: ishitamore1642@gmail.com

Body: hello world

Figure 14. Latest Mail

VIII. Conclusion

The research analyzed helps people with a visual interface to access email in the problematic way, which is the most comprehensive type of contact in the world today. The proposed framework helps decrease the barrier. For example, remember and use the mouse clicks and keyboard shortcuts that were used by the person who visually weakened when receiving the email. The user is solely expected to follow the instructions that will be given by voice technology. In addition to this, the user may need to provide the information via voice entries where mentioned. It plans to develop and implement a real-time e-mail interaction system for people with visual impairments. It is therefore possible to reduce the use of persons with visual impairments by others in their mail-related activities.

IX. Future Scope

Currently, the app only supports commands provided in the English/marathi language and intends to be extended and made available in most languages used in daily life. This allows people from all over the world to access the Web without any problems.

Another requirement of the application can be utilized as future improvements for this project. There are two major disadvantages in this application i.e. accurate speech recognition and the attachment of the image or document. Therefore, going forward, we may add the image or attachment of the document to the sender. Navigation for people with visual impairments through voice instructions and haptic feedback can be added to enhance the existing system. Classification of voice i.e. Male, Female and Child, then recognize the voice using ML Algorithm.

X. Acknowledgment

It is really a pleasure to acknowledge the help and support that has gone in making this thesis. We express our sincere gratitude to our guide Dr. Preetida V-Jani for their invaluable guidance. Without their encouragement this work would not be a reality. With the freedom they provided, we really enjoyed working under them. We would like to mention special thanks to the open source community to provide us with the necessary tools and knowledge base required to complete this project. We thank the Head of Department (HOD) and staff of the Computer Engineering Department for giving us all the facilities required to carry out this research work. We would like to thank all our family members and well wishers for their constant encouragement for all these years, without which we could not have completed this work.

References

[1] Nivedita Bhore, Shraddha Mahala, Komal Acharekar, Dr. Madhavi Waghmare, `Email System for Visually Impaired People ,'International Journal of Engineering Research & Technology (IJERT), 2021

[2] Mullapudi Harshasri, Manyam Durga Bhavani, and Misra Ravikanth, `Voice-Based Email for Blind,' International Journal of Innovative Science and Research Technology, 2021

[3] Paulus A. Tiwari, Pratiksha Zodawan, Harsha P. Nimkar, Trishna Rotke Priya G. Wanjari, `A Review on Voice based E-Mail System for Blind,' International Conference on Inventive Computation Technologies, 2020

[4] Tirthankar Dashgupta, Aakash Anuj, Manjira Sinha, Ritika Gose, Anupam Basu, `VoiceMail Architecture in Desktop and Mobile,' International Conference on Intelligent Human Computer Interactiofn (IHCI), 2012

[5] Harivans Pratap Singh,Aman Pratap Kushwaha, Aayushmaan, Harendra Singh, `Voice Based Email for Blind', International Journal of Scientific Research in Science, Engineering and Technology, , Volume-9, Issue-4, July 2021

[6] Prince Bose;Apurva Malpthak;Utkarsh Bansal;Ashish Harsola, `Digital Assistant For The Blind,' 2nd International Conference for Convergence in Technology (I2CT), 2017

[7] Vinayak Iyer; Kshitij Shah; Sahil Sheth; Kailas Devadkar, `Virtual assistant for the visually impaired,' Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES), 2020

[8] Ezekiel Marvin, `Digital assistant for the blind,' International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020

[9] Carmel Mary Belinda, Rupavathy. N, Mahalakshmi.N.R, `A voice based text mail system for the visually impaired,' International Journal of Engineering & Technology, 2018

[10] Latha L, Babu B, Sowndharya S, `Voice Based Email with Security for Visually Challenged,' International Journal of Engineering and Advanced Technology (IJEAT), 2019