

# Voice Based Email System For Visual Impaired Using AI

Dr . K Badari Nath <sup>1</sup>, Ranjith kumar M <sup>E2</sup>, Nethravathi T <sup>3</sup>, Aman Kumar <sup>4</sup>,

<sup>1</sup>Associate Professor, <sup>2-4</sup>Undergraduate Students, Dept. of Computer science and Engineering, R V college of Engineering, Bangalore, Karnataka, India.

\*\*\*

**Abstract** – E-mail is One of the most popular forms of communication method. Although it is used to interact both professionally and socially, the current Email application is highly challenging for blind users, as they need to remember the keyboard or keypad. The proposed Email system is voice-based, blind users can send and receive emails, and view Proposed system provides a voice based user to access their inbox, sent folder, and trash folder without needing to learn a keyboard or keypad again. Utilizing a speech recognition model helps identify user languages and voices. It allows spoken words to be accurately transcribed into written text, enabling users to interact with their emails using voice commands. The procedure of transferring spoken speech into text is commonly called "speech-to-text" or "automatic speech recognition" (ASR). Google Text-to-Speech (GTTS) technology turns the text that has been inputted into speech. SMTP and IMAP (Internet Message Access Protocol) are two key protocols used in email communication. While SMTP is responsible for sending emails, IMAP is used for accessing and managing email messages. So, that "Blind friendly-Email-System" prototype is developed.

**Key Words** - Speech recognition, speech to text(STT) or "automatic speech recognition" (ASR) , google text to speech(GTTS), and IMAP (Internet Message Access Protocol) .

## 1. INTRODUCTION

Today's population has easy access to any knowledge they choose from the internet. One of the key areas that the Internet has significantly affected is communication. For transmitting and receiving certain crucial information over the Internet, emails are the most dependable mode of communication. However, to access the Internet, people must have a need, which you must be able to see.

In the current email system, those who are visually impaired are unable to read or write emails. Accessing the Internet and the services it offers can be challenging for those with disabilities. Because there is

no other way for blind individuals to use the services on their own, the majority of them are either unable to access or not interested in using the internet's services.

Some solutions to this issue suggest including a third party who is not disabled, Although the level of privacy protection provided by this method may not be the best.

## Objectives

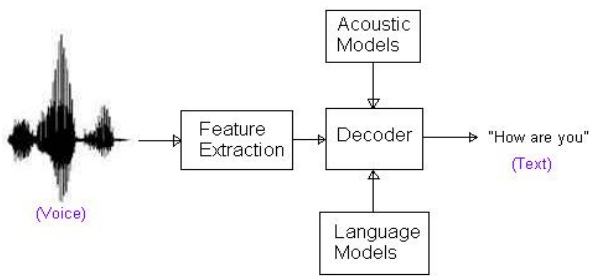
To develop a voice-based emailing system that will enable visually impaired and illiterate persons to use everyday technologies like sending and receiving emails. The proposed system supports E-Mail Compose, Inbox Access and reading/listening mails.

login system, which prevents use of the keyboard and works on the principle of Speech recognition and ported the developed application on a touch enabled standalone system, with the help of designing Blind friendly GUI forms, to access the facilities. So, that "Blind friendly-Email-System" prototype is developed.

## II. SYSTEM ARCHITECTURE AND METHODOLOGY

**Following technologies play a crucial role in the development of the proposed "Blind friendly Email System".**

**Artificial intelligence (AI)** : is the computer-based simulation of human cognitive processes. Expert systems, natural language processing, speech recognition, and machine vision are some specific applications of AI. As shown in the Fig.1, Speech Recognition system has following components.



**Fig.1 Speech Recognition System**

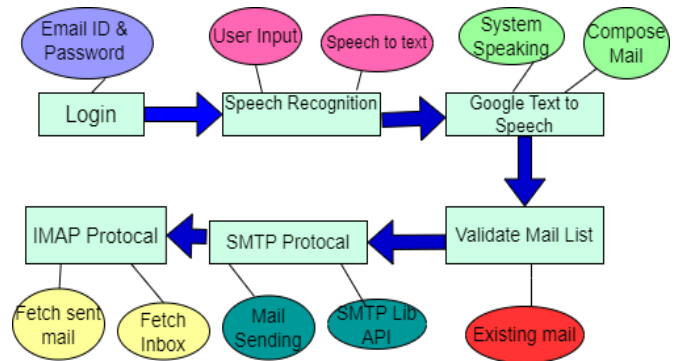
**Training Data:** Trained dataset that contain pairs of audio recordings and their corresponding transcriptions in our project.

**Acoustic modeling:** Speech recognition models employ acoustic modeling techniques to capture the relationship between audio features and phonetic units (such as phonemes or sub-word units). This helps the model understand and differentiate between different sounds in speech.

**Language modeling:** Language modeling is used to enhance the speech recognition process by incorporating knowledge of the specific language being spoken. Language models help the AI technology to provide more precise word predictions or phrases likely to occur based on the context.

**Multiple Language Support:** A large number of languages are supported by Google TTS, including but not limited to English, Spanish, French, German, Italian, Japanese, Korean, Portuguese, Russian, and many more.

Using the above model, our proposed system built and proposed systems, Voice Based Email Model Architecture is shown in the Fig.2.



**Fig.2 Voice Based Email Model Architecture**

In the proposed system, a desktop application is created using python code and many built in libraries and integrated with the Gmail Client. Libraries used and their purpose is illustrated below.

**Speech recognition** library performs speech recognition, converting spoken words into text. It provides an interface to various speech recognition engines and APIs. In speech recognition models, the procedure of converting spoken words into written text is known as **speech-to-text (STT)** conversion. The code snippet used is shown in Fig.3.

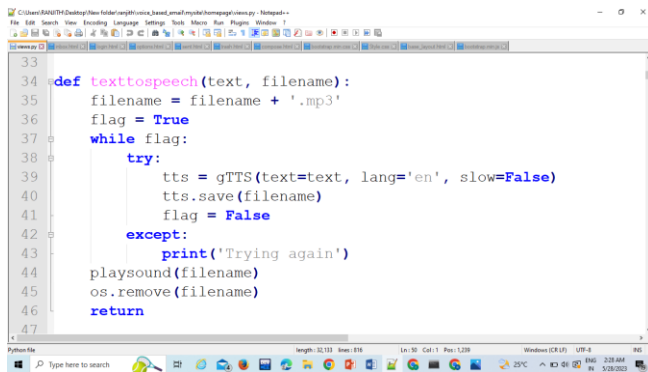
```

47
48 def speechtotext(duration):
49     global i, addr, passwd
50     r = sr.Recognizer()
51     with sr.Microphone() as source:
52         r.adjust_for_ambient_noise(source, duration=1)
53         playsound('speak.mp3')
54         audio = r.listen(source, phrase_time_limit=duration)
55     try:
56         response = r.recognize_google(audio)
57     except:
58         response = 'N'
59     return response
60

```

**Fig.3 Speech to text code**

Google Text-to-Speech library is provided by Google for converting text into speech. As shown in Fig.4, It is possible to use it to create audio files from text. **PyAudio** is a library that permits recording and playing sound streams.



```
33
34 def texttospeech(text, filename):
35     filename = filename + '.mp3'
36     flag = True
37     while flag:
38         try:
39             tts = gTTS(text=text, lang='en', slow=False)
40             tts.save(filename)
41             flag = False
42         except:
43             print('Trying again')
44     playsound(filename)
45     os.remove(filename)
46     return
47
```

**Fig.4 text to Speech code.**

**Imaplib** library is an inherent Python library that empowers getting to and controlling email messages over the IMAP convention. It permits to retrieve messages, search for specific messages, and perform different email-related tasks,

**smtplib** library is another built-in library that provides an interface for sending emails using the Simple Mail Transfer Protocol (SMTP). It allows you to connect to an SMTP server and send emails programmatically.

**Yagmail** is a Python package that simplifies the process of sending emails using Gmail. It provides a higher-level interface for sending emails with Gmail credentials. **playsound** library is a Python library utilized for playing sound records. It gives a basic and cross-stage answer for playing sound records in Python programs.

**request** library is a well known Python library utilized for making HTTP requests and working with web APIs. The "requests" library simplifies the process of sending HTTP requests, handling response data, and interacting with web services. The **httplib2** module is a comprehensive HTTP client library that handles caching, keep-alive, compression, redirects and many kinds of authentication. **pyttsx3** is an offline TTS (Text-to-Speech) library that enables the system to convert text into speech without relying on external services.

**OS** library provides a way to interact with the underlying operating system. It allows us to perform various operating system-related tasks, in our project it helps to check the file existence, access environment variables, and execute system commands, making it

easier to work with the file system and interact with the operating system from within your Python project.

**threading** library allows us to create and manage threads in our Python project. Threads are useful for concurrent execution of multiple tasks, enabling them to perform multiple operations simultaneously or asynchronously. By using threads, it improves the performance of applications, and handles multiple tasks concurrently.

**socket** library in Python provides low-level networking functionality, allowing you to create network connections, send and receive data over different protocols (such as TCP/IP and UDP), and handle network-related tasks. It enables network communication between different machines or processes, making it suitable for implementing client-server applications, network protocols, and networked services. So, in our proposed system using socket library, our application will check the internet condition of the device if the device is connected with the internet or not and give the voice command to the user in the initial stage of the project.

The proposed system focuses on offering fundamental capabilities, including voice-based interaction and the ability to compose, read, send, and receive emails. Proposed system flow chart as indicated in Fig.5, summaries, the different options/facilities provided by the developed application. The proposed application, User Interface module is created using HTML, CSS, js and AJAX modules in python.

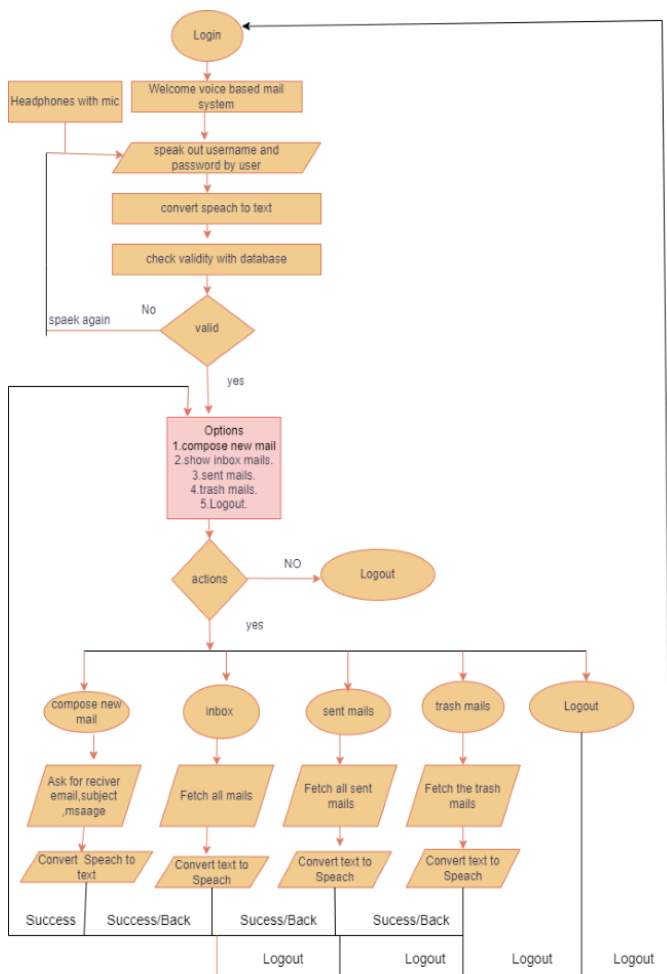


Fig.5 Proposed System Flow Chart

### III. RESULTS AND DISCUSSION

This section presents the details of implementation based on the proposed system.

#### User Login Page

E-mail system login module is altogether voice-based, requiring the user to login with email id and password, Fig 7. At the point when a client talks about their email id and password, the login framework illuminates them and inquires as to whether they are right. Assuming they answer that they will be, they can utilize those subtleties to go on with their activities.

Assuming the client answers that anything is inaccurate, the login framework prompts them again for their email id and password. The front-end for the login page is developed using html,CSS , js and Ajax modules in python. and user email id and password

will be matching with data which is already stored in the db server. We have user sqlite3 for storing and managing the data. The recognized email id and password through speech will be shown in corresponding text boxes.

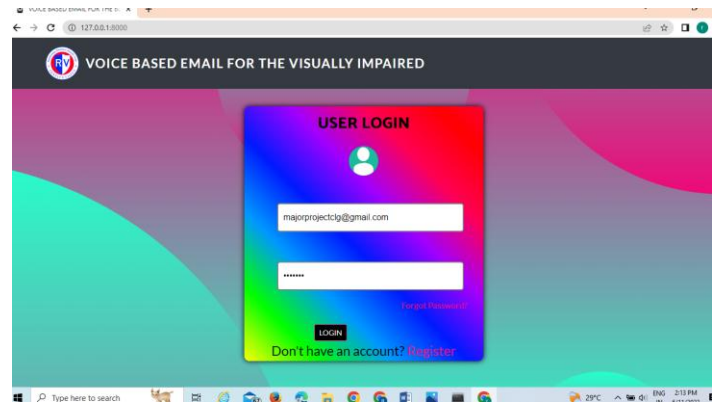


Fig.6 User login page

#### Menu page/option choosing page

Once the login has been successfully completed the menu page will be open the user must click by mouse anywhere on "SCREEN" to perform the operation of mail. The email system prompts the user to choose whether they want to open the compose page, inbox page, sent page or trash page if none of this option users can logout the menu page giving voice command.

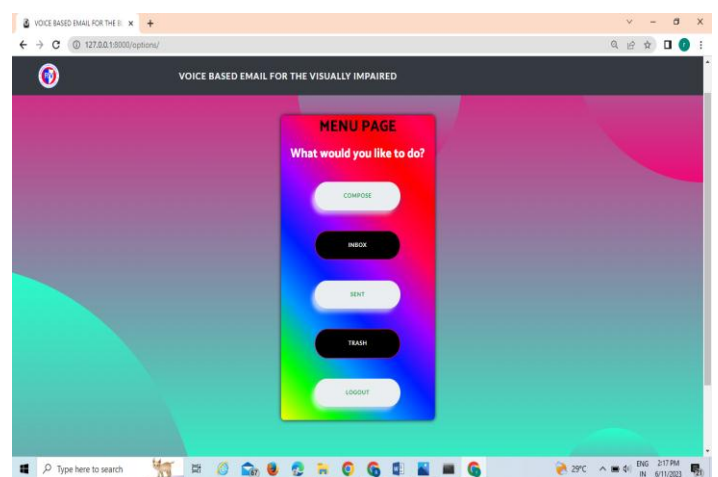


Fig.7 Menu page

### Compose page

This is among the primary options provided by the mail services. The email system prompts the user to compose an email after the option chosen by the user, (Fig.7 and Fig8.). Compose module is entirely voice-based, requiring RECIPIENT'S EMAIL ID to compose mail and also asks for more RECIPIENT'S EMAIL IDs. When a user speaks their RECIPIENT'S EMAIL ID, the system informs them and asks if they are correct. If the user responds that they are, they can use those details to continue with their operations. If the user replies that is incorrect, the compose system prompts them once more for their RECIPIENT'S EMAIL ID. Once this confirmed system prompts the user to speak SUBJECT and BODY messages. Once the user completes the subject and body message system informs the user to add attachment. Attachment includes audio record or file name with extension such as jpg, png or pdf. The recognized receiver email, subject and body of the message through speech is shown in corresponding text boxes.

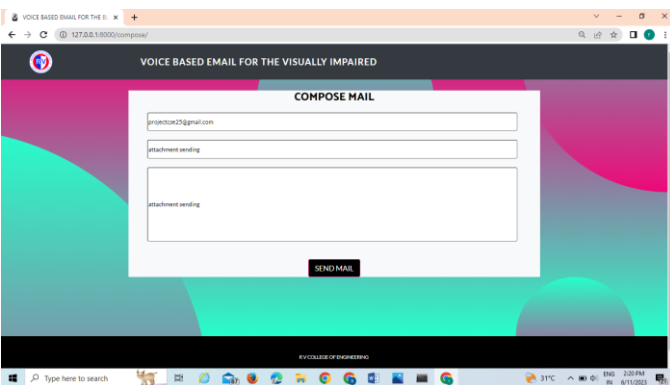


Fig.8 Composing mail at sender side

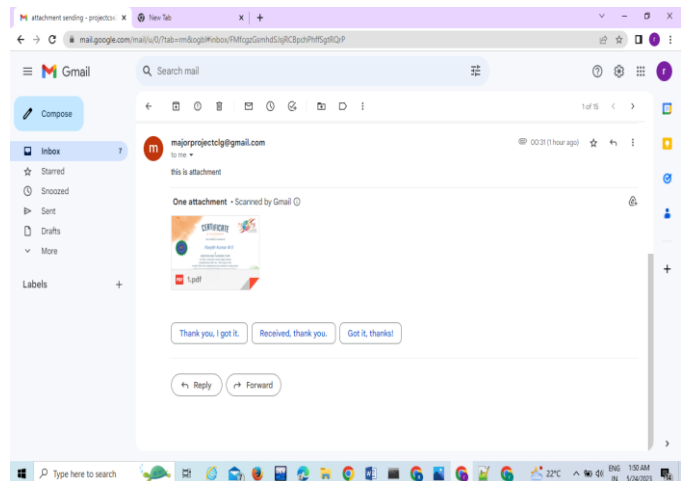


Fig.9 Acknowledgement at receiver side

### Inbox page

In the Inbox page users listen to the unread emails, reply to the particular email, forwarding the email to others by performing the mouse click operation and users can start doing operation Fig.8 and Fig.9. The user is initially prompted to identify the precise user whose communication has to be read. The machine then reads the message's subject contents. The Google Text-to-Speech (gTTs) reading method is employed. If they want to read the specific email by choosing the search option. If the user chooses the unread option then the system reads the message's subject content. After that asks the user to perform the reply, the forward or back to the menu page. Back option is used to come back to the menu page.

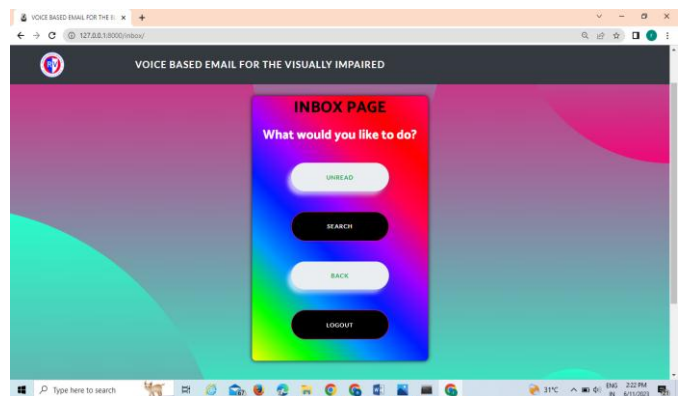


Fig.10 Inbox page

The Unread Mails in the user mail in our proposed system is read mail via voice. We can see in the command prompt it will show the read mail. Shown in Fig.11

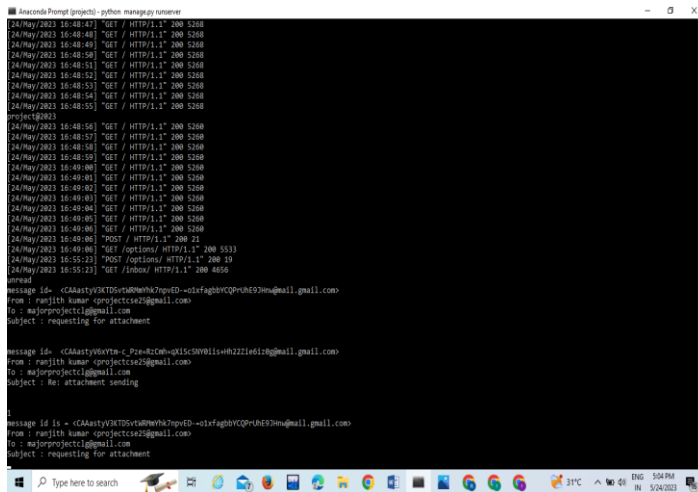


Fig.11 First three Unread messages

### Sent page

This module is used to listen to sent mails. This module includes two options: search and back. Search option is used to search the sent mail by using a specific email id, our application takes input from the user according to the input application will do. Back option helps to go back to the menu page. Shown in Fig.10

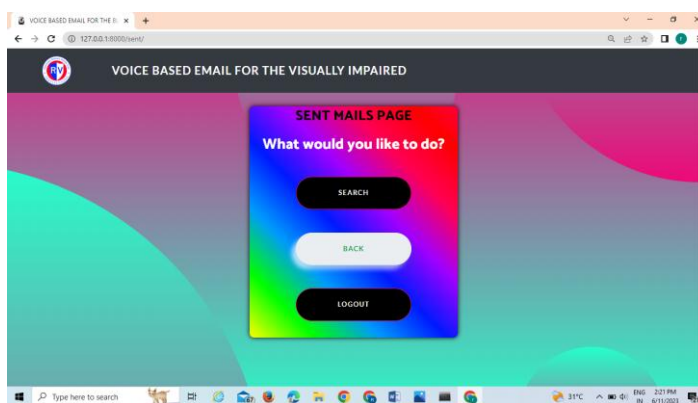


Fig.12 Sent mails page

### Trash page

This module is used to listen to the deleted mails, Fig.11. The deleted mails can be listened to by using a specific mail id.

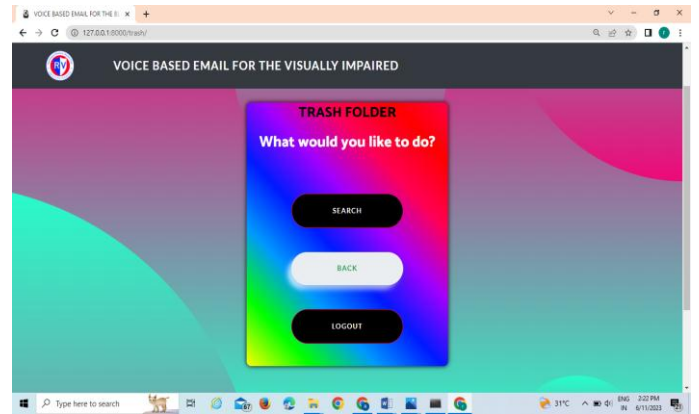


Fig.13 Trash page

## VI. CONCLUSION

This web application is developed by our team keeping in mind the difficulties of the visually impaired, helping them to use email services. Screen readers have been removed. This makes the system much more compatible, physically challenged individuals will be able to interact with the outside world. Regardless of age, everyone can use the proposed email system with ease. It has speech-to-content features as well as content-to-speech features. With this, people who are blind or visually impaired can send and receive mail easily with just voice commands and very minimal mouse usage. The current developed system only functions on desktop computers. In the future, there is potential to include this facility as an application in mobile phones since their use is now on the rise.

### REFERENCES

- [1]. Abhiram J , Amrutha S , Aneetta Susan John3, Jinshu Maria John , Midhun V Nair, Alpha Mathew, May 2022, Voice Based Email for Visually Challenged, International Journal for Research in Applied Science & Engineering Technology (IJRASET).
- [2]. Sanjeevi Kumar P, Ashwitha Shetty, Megha Manjunath Naik, Nayak Ashmitha Suresh, Sachin, May-2022, voice based email system for visually impaired, International Research Journal of Modernization in Engineering Technology and Science.

[3]. Mullapudi Harshasri, Manyam Durga Bhavani, and Misra Ravikanth, July 2021, Voice Based Email for Blind, International Journal of Innovative Research.

[4]. Jain, Viram and A K, Krithika and Shenoy, Rachana N and A, Rakshitha and V, Suma and Ahmed, Muzameel, Voice Based Email for the Visually Impaired (July 9, 2021). Proceedings of the International Conference on IoT Based Control Networks & Intelligent Systems - ICICNIS 2021, [ssrn.com/abstract=3883411](https://ssrn.com/abstract=3883411).

[5]. Parkhi Bhardwaj, & Gunjan Sethi (2020) Voice Based E-mail System for Visually Impaired. International Research Journal of Engineering and Technology (IRJET).

[6]. Mamatha, A., Jade, V., Saravana, J., Purshotham, A., & Suhas, A. V. (2020). Voice Based E-mail System for Visually Impaired. International Journal of Research in Engineering, Science and Management, 3(8), 51- 54.

[7]. "A Review on voice based email system for blind" 09 June 2020. Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT-2020) IEEE Xplore Part Number: CFP20F70-ART; ISBN: 978-1-7281-4685-0.

[8]. Paulus A. Tiwari, Pratiksha Zodwan, Harsha P. Nimkar, Trishna Rotke, Priya G. Wanjari, Umesh Saarth "A Review on voice based email system for blind" 09 June 2020. Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT-2020) IEEE Xplore Part Number: CFP20F70-ART; ISBN: 978-1-7281-4685-0.

[9]. Sherly Noel, October 2020, "Human computer interaction (HCI) based smart voice email (Vmail) application assistant for visually impaired users (VIU)" Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICCSIT 2020).

[10]. Pathan, N., Bhojar, N., Lakra, U., & Lilhare, D. (2019). V-Mail (Voice Based E-Mail Application).

[11]. Milan Badigar, Nikita Dias, Jemima Dias and Mario Pinto, "Voice Based Email Application For Visually

Impaired. International Journal of Science Technology & Engineering - Volume 4, Issue 12, June 2018, pp. 166-170.

[12]. Rastogi R., Mittal S., Aggarwal S., CSE Dept., ABES Engineering College - "A novel approach for communication among blind, deaf and dumb people", November 2018, IEEE.

### About the Authors



Dr. Badari Nath K

Associate professor, Dept. of Computer Science & Engg  
R V College of Engineering  
Bangalore, India.



Bachelor of Engineering, Final Year student seeking his degree in Computer Science & Engineering at R V College of Engineering, Bangalore, Karnataka, India.

E-mail: [ranjith15035@gmail.com](mailto:ranjith15035@gmail.com)



Bachelor of Engineering, Final Year student seeking her degree in Computer Science & Engineering at R V College of Engineering, Bangalore, Karnataka, India.

E-mail: [nethravathit.cs19@rvce.edu.in](mailto:nethravathit.cs19@rvce.edu.in)



Bachelor of Engineering, Final Year student seeking his degree in Computer Science & Engineering at R V College of Engineering, Bangalore, Karnataka, India.

E-mail: [amankumar.cs19@rvce.edu.in](mailto:amankumar.cs19@rvce.edu.in)