



**VOICE BASED EMAIL SYSTEM FOR BLIND**

**Project Progress Report**



**SUBMITTED IN PARTIAL FULFILLMENT OF THE**

**REQUIREMENT FOR THE AWARD OF THE DEGREE OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

***Submitted by:-***



|  |  |
| --- | --- |
| Name | Univ. Roll No. |
| 1. Raj Shekhar | 20033445025 |
| 1. Ranjeet Kumar Mahto | 20033445027 |
| 1. Prem Kumar | 20033445018 |
| 1. Sweta Kumari | 20033445037 |
| 1. Priyanka Kumari | 20033445020 |

**Under the Guidance of**

Mrs. Shreosee Bhattacharya

**Assistant Professor, Dept. of CSE**



**CERTIFICATE OF RECOMMENDATION**

This is to recommend that the work undertaken in this Project Report entitled **“ Voice Based Email for Blind ”** has been carried out by Raj Shekhar (20033445025), Ranjeet Kumar Mahto (20033445027), Prem Kumar (20033445018), Sweta Kumari (2033445037), Priyanka Kumari (20033445020) under my supervision and may be accepted in partial fulfilment of the requirement for the degree of Bachelor of Technology in Computer Science Engineering.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Counter signed by,**

**Mr. Ashim Kumar Mahato Head of the Department**

**Department of CSE**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assistant Professor,**

**Mrs. Shreosee Bhattacharya**

**Department of CSE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner**

# 

# CERTIFICATION OF APPROVAL

# The foregoing Thesis I hereby approved as a creditable study of engineering subject to warrant its acceptance as a pre-requisite to obtain the degree for which it has been submitted. It is understood that by this approval the undersigned don’t necessarily endorse or approve any statement made, opinion or conclusion drawn there in but approve the thesis only for the purpose for which it is submitted

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of the Dept.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mentor**

**Date: - \_\_\_\_\_\_\_\_\_\_\_**

**Place: - \_\_\_\_\_\_\_\_\_\_\_**



# DECLARATION

I thereby declare that the project report entitled “**Voice Based Email for Blind**” submitted by us to "RAMGARH ENGINEERING COLLEGE" is our original work and has not been submitted to any other institution for any purpose

I affirm that the project was carried Out under the guidance of "SHREOSEE MADAM AND ASHIM SIR" and that the work presented in the report is authentic, accurate and complete to the best of our knowledge.

I also declare that we have followed all ethical guidelines and have not engaged in any form of academic misconduct and used our own ideas.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Raj Shekhar**

**(20033445025)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Ranjeet Kumar Mahto Prem Kumar**

**(20033445027) (20033445018)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Sweta Kumari Priyanka Kumari**

**(2033445037) (20033445020)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner Name**

# 

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner Signature**

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 

# UNDERTAKING

We, Raj Shekhar (20033445025), Ranjeet Kumar Mahto (20033445027), Prem Kumar ( 2003344501), Sweta Kumari (2033445037), Priyanka Kumari (2003344501), of B.Tech (Computer Science & Engineering) hereby declare that our report, titled “ **Voice Based Email for Blind** ” is our own contribution. We declare that we have not indulged in any form of plagiarism while writing this report. The work or ideas of other authors which are utilized in this report has been properly acknowledged and mentioned in the Bibliography. We undertake total responsibility if any trace of plagiarism is found at any later stage. In such a case my degree will be cancelled automatically.

|  |  |
| --- | --- |
| Name | Univ. Roll No. |
| 1. Raj Shekhar | 20033445025 |
| 1. Ranjeet Kumar Mahto | 20033445027 |
| 1. Prem Kumar | 20033445018 |
| 1. Sweta Kumari | 20033445037 |
| 1. Priyanka Kumari | 20033445020 |

# 

# ACKNOWLEDGMENT

# A Study of this volume can never be the outcome of single person or just a group of dedicated students, so we express our gratitude to those who helped and supported us in completing our project because successful completion of any work requires guidance and help from a no of people

# Firstly, I would like to thanks REC to provide us the opportunity to do such a project. We are very much thankful to our H.O.D. Mr. Ashim Kumar Mahato sir, for his uninterrupted support in development of project and have enabled us to complete the project o given period of time…. They were always there to listen and to give advice and taught us how to express our ideas & views professionally.

# Last but not least I want to acknowledge all our friends and colleagues who supported us in our project.

# 

# ABSTRACT

Internet become a part and parcel in our day -to- day life but as per our title blind face challenges. The gradual advancement of computer based accessible systems has manifold aspects for the visually impaired. Virtual environment based on audio such as the screen readers shaped blind to access internet very much. Voicemail system architecture can be used by a blind to access e-mails with full of efficiency. The purpose of the project has enabled the blind not only to send but also to receive messages. The implementation of voice to text and text to voice technique has accessed for blind. The system is completely based on interactive voice response which will make it efficient for the visually impaired.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
|  | **Abstract** | **7** |
| **1.** | **Introduction** | **10** |
|  | 1.1 Interactive Voice Response | 10 |
|  | 1.2 Speech Recognition | 11 |
|  | 1.3 Speech To Text | 11 |
|  | 1.4 Text To Speech | 12 |
|  | 1.5 Purpose of The Project | 12 |
|  | 1.6 Motivation | 13 |
| **2.** | **Literature Review** | **14** |
|  | 2.1 Voice Based System in Desktop and Mobile Devices for Blind People | 14 |
|  | 2.2 Voice Based Search Engine and Web page Reader | 15 |
|  | 2.3 Voice Based Services for Blind People | 16 |
|  | 2.4 Voice based e-mail System for Blinds | 19 |
| **3.** | **System Development** | **21** |
|  | 3.1 Existing System | 21 |
|  | 3.2 Proposed System | 22 |
|  | 3.3 Component Description | 22 |
|  | 3.3.1 Detail Description of IVR | 22 |
|  | 3.3.2 Working of Speech Recognition | 24 |
|  | 3.3.3 Speech To Text Converter | 26 |
|  | 3.3.4 Speech Synthesis | 27 |
| **4.** | **Design** | **30** |
|  | 4.1 Design Phases of The Proposed System | 30 |
| **5.** | **Methodology** | 33 |
| **6.** | **Implementation** | **35** |
|  | 6.1 Speech Recognition in Python | 35 |
|  | Required Installations | 36 |
|  | Text to Speech in Python | 37 |
|  | gTTS | 37 |
|  | 6.2 System Requirements | 38 |
|  | 6.3 Hardware Requirements | 38 |

|  |  |  |
| --- | --- | --- |
| **7.** | **Testing And Results** | **39** |
|  | 7.1 Code | 39 |
|  | 7.2 Screenshot | 50 |
| **8.** | **Conclusion** | **53** |
|  | 8.1 Future Scope | 54 |
|  | 8.2 Advantages | 54 |
|  | **References** | 55 |

**Chapter 1**

**INTRODUCTION**

We have seen that the introduction of Internet has revolutionized many fields. Internet has made life of people so easy that people today have access to any information they want easily. Communication is one of the main fields highly changed by Internet.

E-mails are the most dependable way of communication over Internet, for sending and receiving some important information. But there is a certain norm for humans to access the Internet and the norm is you must be able to see. But there are also differently able people in our society who are not gifted with what you have. There are some visually impaired people or blind people who can’t see things and thus can’t see the computer screen or keyboard.

A survey has shown that there are more than 240 million visually impaired people around the globe. That is, around 240 million people are unaware of how to use Internet or E-mail. The only way by which a visually challenged person can send an E-mail is, they have to speak the entire content of the mail to another person (not visually challenged) and then that third person will compose the mail and send on the behalf of the visually challenged person. But this is not a right way to deal with the problem. It is very unlikely that every time a visually impaired person can find someone for help. Although for these reasons the visually impaired people are criticized by our society.

So, for the betterment of society and giving an equal status to such specially able people we have come up with this project idea which provides the user with ability to send mails using voice commands without the need of keyboard or any other visual things.

# INTERACTIVE VOICE RESPONSE

Interactive voice response (IVR) is a technology that allows a computer to interact with humans through the use of voice and DTMF tones input via a keypad.

In telecommunications, IVR allows customers to interact with a company’s host system via a telephone keypad or by speech recognition, after which services can be inquired about through the IVR dialogue. IVR systems can respond with pre-recorded or dynamically generated audio to further direct users on how to proceed.

# SPEECH RECOGNITION

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT).

# SPEECH TO TEXT

# 

The system acquires speech at run time through a microphone and processes the sampled speech to recognize the uttered text. The recognized text can be stored in a file. We are developing this on Android platform using Eclipse workbench. Our speech to-text system directly acquires and converts speech to text. It can supplement other larger systems, giving users a different choice for data entry. A speech-to-text system can also improve system accessibility by providing data entry options for blind, deaf, or physically handicapped users. Speech recognition system can be divided into several blocks: feature extraction, acoustic models database which is built based on the training data, dictionary, language model and the speech recognition algorithm. Analog speech signal must first be sampled at time and amplitude axes, or digitized. Samples of the speech signal are analyzed in even intervals. This period is usually 20 ms because the signal in this interval is considered stationary. Speech feature extraction involves the formation of equally spaced discrete vectors of speech characteristics. Feature vectors from training database are used to estimate the parameters of acoustic models. The acoustic model describes properties of the basic elements that can be recognized. The basic element can be a phoneme for continuous speech or word for isolated words recognition.

# TEXT TO SPEECH

Converting text to voice output using speech synthesis techniques. Although initially used by the blind to listen to written material, it is now used extensively to convey financial data, e-mail messages, and other information via telephone for everyone. Text-to-speech is also used on handheld devices such as portable GPS units to announce street names when giving directions. Our Text-to-Speech Converter accepts a string of 50 characters of text (alphabets and/or numbers) as input. In this, we have interfaced the keyboard with the controller and defined all the alphabets as well as digits keys on it. The speech processor has an unlimited dictionary and can speak out almost any text provided at the input most of the times. Hence, it has an accuracy of above 90%. It is a microcontroller based hardware coded in Embedded C language. Further research is to be done to optimize various methods of inputting the text i.e. Reading the text using optical sensor and converting it to speech so that almost all sorts of physical challenges faced by the people while communicating are overcome.

# PURPOSE OF THE PROJECT

This project proposes a python based application, designed specifically for visually impaired people. This application provide a voice based mailing service where they could read and send mail on their own, without any guidance through their Email accounts. The VMAIL system can be used by a blind person to access mails easily and adeptly. Hence dependence of visually challenged on other individual for their activities associated to mail can be condensed.

The application will be a python-based application for visually challenged persons using IVR- Interactive voice response, thus sanctioning everyone to control their mail accounts using their voice only and to be able to read, send, and perform all the other useful tasks. The system will ask the user with voice commands to perform certain action and the user will respond to it. The main advantage of this system is that use of keyboard is completely eliminated, the user will have to respond through voice only.

# MOTIVATION

It is estimated that nearly 285 million people in the world are visually impaired and idea is to facilitate suitable communication system for them. This reason was driving force behind developing given system. One of the major disadvantages of existing system is that all operations are based on mouse click events and keyboard. Operations depend completely on types of clicks specified by idea. Also sometimes remembering keyboard shortcut is difficult. The extent of existing system is limited for blind and visually impaired people. There is high need of developing a proper system which curbs all the above drawbacks and turn into a simple system. Idea focuses on providing basic functionalities like compose, send, receive E-mail along with advance features like voice based operation, search mail, provision for voice as well as text based email with added ease and simplicity. Related Work Interaction of the users to the system earlier was based on Screen reader based technology and also system based on mouse click based operations were in for every operation there is associated mouse click for example to compose email let say to left clicks. Therefore interaction with the system is tough also there is need to keep events in mind. This paper focuses on developing an email system which helps blind people to use communication services. The system based in IVR is used, major idea is to discard keyboard and use of mouse operation. Internet is rich source of knowledge and information, blind people face difficulties in accessing text based material. The idea is to develop audio feedback based virtual environment like screen reader, text to speech, etc. Voice mail architecture helps blind people to access info. in form of audio, text, self-read system. Idea focuses on helping visually impaired and illiterate people to access technology by reducing cognitive load. Decision making depends on eyesight and everything that happens or appears.

**Chapter 2**

**LITERATURE REVIEW**

* 1. **“Voice Based System in Desktop and Mobile Devices for Blind People”. In International Journal of Emerging Technology and Advanced Engineering (IJETAE), 2014**

This paper deals with “Voice Based System in Desktop and Mobile Devices for Blind People”. Voice mail architecture helps blind people to access e-mail and other multimedia functions of operating system (songs, text).Also in mobile application SMS can be read by system itself. Now a days the advancement made in computer technology opened platforms for visually impaired people across the world. It has been observed that nearly about 60% of total blind population across the world is present in INDIA. In this paper, we describe the voice mail architecture used by blind people to access E-mail and multimedia functions of operating system easily and efficiently. This architecture will also reduce cognitive load taken by blind to remember and type characters using keyboard. There is bulk of information available on technological advances for visually impaired people. This includes development of text to Braille systems, screen magnifiers and screen readers. Recently, attempts have been made in order to develop tools and technologies to help Blind people to access internet technologies. Among the early attempts, voice input and input for surfing was adopted for the Blind people. In IBM’s Home page the web page is an easy-to-use interface and converts the text-to-speech having different gender voices for reading texts and links. However, the disadvantage of this is that the developer has to design a complex new interface for the complex graphical web pages to be browsed and for the screen reader to recognize.

Simple browsing solution, which divides a web page into two dimensions. This greatly

simplifies a web page’s structure and makes it easier to browse. Another web browser

generated a tree structure from the HTML document through analysing links. As it attempted to structure the pages that are linked together to enhance navigability, it did not prove very efficient for surfing. After, it did not handle needs regarding navigability and usability of current page itself. Another browser developed for the visually handicapped people was guided which had an integrated TTS engine. This system applies some advanced text extraction algorithm to represent the page in a user-friendly manner. However, still it did not meet the required standards of commercial use. Considering Indian scenario, Shruti Drishti and Web Browser for Blind are the two web browser framework that are used by Blind people to access the internet including the emails. Both the systems are integrated with Indian language ASR and TTS systems. But the available systems are not portable for small devices like mobile phones.

# “Voice Based Search Engine and Web page Reader”. In International Journal of

**Computational Engineering Research (IJCER)**

This paper aims to develop a search engine which supports Man-Machine interaction purely in the form of voice. A novel Voice based Search Engine and Web-page Reader which allows the users to command and control the web browser through their voice, is introduced. The existing Search Engines get request from the user in the form of text and respond by retrieving the relevant documents from the server and displays in the form of text .Even though the existing web browsers are capable of playing audios and videos, the user has to request by typing some text in the search text box and then the user can play the interested audio/video with the help of Graphical User Interfaces (GUI). The proposed Voice based Search Engine aspires to serve the users especially the blind in browsing the Internet. The user can speak with the computer and the computer will respond to the user in the form of voice. The computer will assist the user in reading the documents as well. Voice-enabled interface with addition support for gesture based input and output approaches are for the “Social Robot Maggie” converting it into an aloud reader . This voice recognition and synthesis can be affected by number of reasons such as the voice pitch, its speed, its volume etc. It is based on the Loquendo ETTS (Emotional Text-To-Speech) software. Robot also expresses its mood through gesture that is based on gestionary. Speech recognition accuracy can be improved by removal of noise. In A Bayesian scheme is applied in a wavelet domain to separate the speech and noise components in a proposed iterative speech enhancement algorithm. This proposed method is developed in the wavelet domain to exploit the selected features in the time frequency space representation. It involves two stages: a noise estimate stage and a signal separation stage. In the Principle Component Analysis (PCA) based HMM for the visual modality of audio-visual recordings is used. PCA (Principle Component Analysis) and PDF (Probabilistic Density Analysis). Presents an approach to speech recognition using fuzzy modelling and decision making that ignores noise instead of its detection and removal. In the speech spectrogram is converted into a fuzzy linguistic description and this description is used instead of precise acoustic features. In Voice recognition technique combined with facial feature interaction to assist virtual artist with upper limb disabilities to create visual cut in a digital medium, preserve the individuality and authenticity of the art work. Techniques to recover phenomena such as Sentence Boundaries, Filler words and Disfluencies referred to as structural Metadata are discussed in and describe the approach that automatically adds information about the location of sentence boundaries and speech disfluencies in order to enrich speech recognition output. Clarissa a voice enabled procedure browser that is deployed on the international space station (ISS). The main components of the Clarissa system are speech recognition module a classifier for executing the open microphone accepts/reject decision, a semantic analysis and a dialog manager. Mainly focuses on expressions. To build a prosody model for each expressive state, an end pitch and a delta pitch for each syllable are predicted from a set of features gathered from the text. The expression- tagged units are then pooled with the neutral data, In a TTS system, such paralinguistic events efficiently provide clues as to the state of a transaction, and Markup specifying these events is a convenient way for a developer to achieve these types of events in the audio coming from the TTS engine.

Main features of are smooth and natural sounding speech can be synthesized, the voice characteristics can be changed, it is “trainable. Limitations of the basic system is that synthesized speech is “buzz” since it is based on a vocoding technique, it has been overcome by high quality vocoder and hidden semi-Markov model based acoustic modelling. Speech synthesis consists of three categories: Concatenation Synthesis, Articulation Synthesis, and Formant Synthesis.

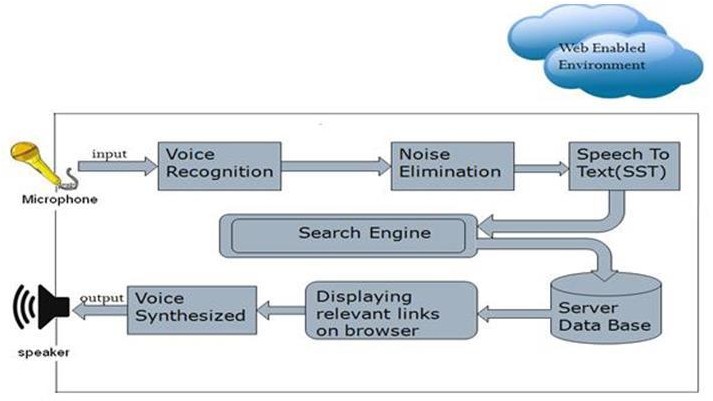
Mainly focuses on formant synthesis, array of phoneme of syllable with formants frequency is given as input, frequency of given input is processed, on collaborated with Thai-Tonal-Accent Rules convert given formants frequency format to wave format, so that audio output via soundcard.

Figure-1: Voice Recognition Flow Diagram

# “Voice Based Services for Blind People”. In International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT)

The advancement in computer based accessible systems has opened up many avenues for the visually impaired across a wide majority of the globe. Audio feedback based virtual environment like, the screen readers have helped blind people to access internet applications immensely. However, a large section of visually impaired people in different countries, in particular, the Indian sub-continent could not benefit much from such systems. This was primarily due to the difference in the technology required for Indian languages compared to those corresponding to other popular languages of the world. In this paper, we describe the voicemail system architecture that can be used by a blind person to access e-mails easily and efficiently. The contribution made by this research has enabled the blind people to send and receive voice-based e-mail messages in their native language with the help of a mobile device. Our proposed system GUI has been evaluated against the GUI of a traditional mail server. We found that our proposed architecture performs much better than that of the existing GUIS. In this project, we use voice to text and text to voice technique access for blind people.

The navigation system uses TTS (Text-to-Speech) for blindness in order to provide a navigation service through voice. Suggested system, as an independent program, is fairly cheap and it is possible to install onto Smartphone held by blind people. This allows blind people to easy access the program. An increasing number of studies have used technology to help blind people to integrate more fully into a global world. We present software to use mobile devices by blind users. The software considers a system of instant messenger to Favor interaction of blind users with any other user connected to the network. Nowadays the advancement made in computer technology opened platforms for visually impaired people across the world. It has been observed that nearly about 60% of the total blind population across the world is present in INDIA. In this paper, we describe the voice mail architecture used by blind people to access E-mail and multimedia functions of the operating system easily and efficiently. This architecture will also reduce cognitive load taken by the blind to remember and type characters using the keyboard. It also helps handicapped and illiterate people. In previous work, blind people does not send email using the system. The multitude of email types along with the ability setting enables their use in nomadic daily contexts. But these emails are not useful in all types of people such as blind people they can’t send the email. Audio based email are only preferable for blind peoples. They can easily respond to the audio instructions. In this system is very rare. So there is less chance to available this audio based email to the blind people. We describe the voicemail system architecture that can be used by a blind person to access e-mails easily and efficiently. The contribution made by this research has enabled the blind people to send and receive voice-based e-mail messages in their native language with the help of a computer or a mobile device. Our proposed system GUI has been evaluated against the GUI of a traditional mail server. We found that our proposed architecture performs much better than that of the existing GUIS.

It involves the development of the following modules:

**SPEECH\_ TO\_ TEXT Converter** :The system acquires speech at run time through a microphone and processes the sampled speech to recognize the uttered text. The recognized text can be stored in a file. We are developing this on Android platform using Eclipse workbench. Our speech to-text system directly acquires and converts speech to text. It can supplement other larger systems, giving users a different choice for data entry. A speech-to-text system can also improve system accessibility by providing data entry options for blind, deaf, or physically handicapped users. Speech recognition system can be divided into several blocks: feature extraction, acoustic models database which is built based on the training data, dictionary, language model and the speech recognition algorithm. Analog speech signal must first be sampled at time and amplitude axes, or digitized. Samples of the speech signal are analysed in even intervals. This period is usually 20 ms because the signal in this interval is considered stationary. Speech feature extraction involves the formation of equally spaced discrete vectors of speech characteristics. Feature vectors from training database are used to estimate the parameters of acoustic models. The acoustic model describes properties of the basic elements that can be recognized. The basic element can be a phoneme for continuous speech or word for isolated

words recognition.

**TEXT\_ TO\_ SPEECH Converter:** Converting text to voice output using speech synthesis techniques. Although initially used by the blind to listen to written material, it is now used extensively to convey financial data, e-mail messages, and other information via telephone for everyone. Text-to-speech is also used on handheld devices such as portable GPS units to announce street names when giving directions. Our Text-to- Speech Converter accepts a string of 50 characters of text (alphabets and/or numbers) as input. In this, we have interfaced the keyboard with the controller and defined all the alphabets as well as digits keys on it. The speech processor has an unlimited dictionary and can speak out almost any text provided at the input most of the times. Hence, it has an accuracy of above 90%. It is a microcontroller based hardware coded in Embedded C language. Further research is to be done to optimize various methods of inputting the text i.e. Reading the text using optical sensor and converting it to speech so that almost all sorts of physical challenges faced by the people while communicating are overcome. WORD RECOGNITION :Voice recognition software (also known as speech to text software)allows an individual to use their voice instead of typing on a keyboard. Voice recognition may be used to dictate text into the computer or to give commands to the computer. Voice recognition software allows for a quick method of writing onto a computer. It is also useful for people with disabilities who find it difficult to use the keyboard. This software can also assist those who have difficulty with transferring ideas onto paper as it helps take the focus out of the mechanics of writing. Word recognition is measured as a matter of speed, such that a word with a high level of recognition is read faster than a novel one. This manner of testing suggests that comprehension

of the meaning of the words being read is not required, but rather the ability to recognize them in a way that allows proper pronunciation. Therefore, context is unimportant, and word recognition is often assessed with words presented in isolation in formats such as flash cards Nevertheless, ease in word recognition, as in fluency, enables proficiency that fosters comprehension of the text being read

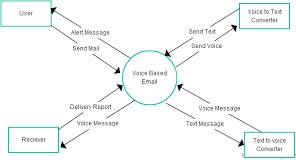


Figure-2: System Data Flow Diagram

# “Voice based e-mail System for Blinds”. In International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)

Internet plays a vital role in today’s world of communication. Today the world is running on the basis of internet. No work can be done without use of internet. Electronic mail i.e. email is the most important part in day to day life. But some of the people in today’s world don’t know how to make use of internet, some are blind or some are illiterate. So it goes very difficult to them when to live in this world of internet. Nowadays there are various technologies available in this world like screen readers, ASR, TTS, STT, etc. but these are not that much efficient for them. Around 39 million people are blind and 246 people have low vision and also 82 of people living with blindness are 50 aged and above. We have to make some internet facilities to them so they can use internet. Therefore we came up with our project as voice based email system for blinds which will help a lot to visually impaired peoples and also illiterate peoples for sending their mails. The users of this system don’t need to remember any basic information about keyboard shortcuts as well as location of the keys. Simple mouse click operations are needed for functions making system easy to use for user of any age group. Our system provides location of where user is prompting through voice so that user doesn’t have to worry about remembering which mouse click operation

The visually challenged people find it very difficult to utilize this technology because of the fact that using them requires visual perception. However not all people can use the internet. This is because in order to access the internet you would need to know what is written on the screen. If that is not visible it is of no use. This makes internet a completely useless technology for the visually impaired and illiterate people.

**In this system mainly three types of technologies are used namely:**

**STT (Speech-to-text):** here whatever we speak is converted to text. Their will a small icon ofmic on whose clicking the user had to speak and his/her speech will be converted to text format, which the naked people would see and read also.

**TTS (text-to-speech)** this, method is full opposite of STT. In this method, which converts the text format of the emails to synthesized speech? A text-to-speech (TTS) system converts language text into speech, alternative systems render symbolic linguistic representations. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database.

**IVR (Interactive voice response**): IVR is an advanced technology describes the interaction between the user and the system in the way of responding by using keyboard for the respective voice message. IVR allows user to interact with an email host system

via a system keyboard, after that users can easily service their own enquiries by listening to the IVR dialogue. IVR systems generally respond with pre-recorded audio voice to further assist users on how to proceed.

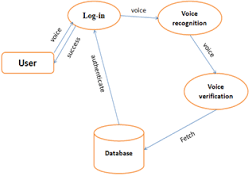
The audio that would be pre-recorded and the system need to have large volumes.

Figure-3: Proposed System Architecture

**Chapter 3**

**SYSTEM DEVELOPMENT**

* 1. **EXISTING SYSTEM**

There are a complete number of 4.1 billion email accounts made until 2014 and a there will be evaluated 5.2 billion records by end of 2018. This makes messages the most utilized type of correspondence. The most generally perceived mail benefits that we use in our regular day to day existence can't be used by ostensibly tried people. This is on the grounds that they don't give any office so the individual in front can hear out the substance of the screen. As they can't imagine what is now present on screen they can't make out where to click so as to play out the necessary tasks. For an outwardly tested individual utilizing a PC just because isn't that helpful for what it's worth for an ordinary client despite the fact that it is easy to understand. In spite of the fact that there are many screen readers accessible then likewise these individuals face some minor troubles.

Screen readers read out whatever substance is there on the screen and to play out those activities the individual should utilize console alternate routes as mouse area can't be followed by the screen reader. This implies two things; one that the client can't utilize mouse pointer as it is totally awkward if the pointer area can't be followed and second that client ought to be knowledgeable with the console concerning where every single key is found. A client is new to PC can accordingly not utilize this administration as they don't know about the key areas. Another disadvantage that sets in is that screen reader read out the substance in successive way and subsequently client can make out the substance of the screen just on the off chance that they are in essential HTML position.

Therefore the new propelled pages which don't follow this worldview so as to make the site more easy to use just make additional issues for these individuals. Moreover the systems that do use only voice for interaction between the user and the system don’t have good voice transcription. All these are a few downsides of the present framework which we will defeat in the framework we are creating.

# PROPOSED SYSTEM

The proposed system is based on a completely novel idea and is nowhere like the existing mail systems. The most important aspect that has been kept in mind while developing the proposed system is accessibility. A web system is said to be perfectly accessible only if it can be used efficiently by all types of people whether able or disable. The current systems do not provide this accessibility. Thus the system we are developing is completely different from the current system. Unlike current system which emphasizes more on user friendliness of normal users, our system focuses more on user friendliness of all types of people including normal people visually impaired people as well as illiterate people. The complete system is based on IVR- interactive voice response.

When using this system the computer will be prompting the user to perform specific operations to avail respective services and if the user needs to access the respective services then he/she needs to perform that operation. One of the major advantages of this system is that user won’t require to use the keyboard. All operations will be based on voice commands. This system will be perfectly accessible to all types of users as it is just based on simple speech inputs and there is no need to remember keyboard shortcuts. Also because of IVR facility those who cannot read need not worry as they can listen to the prompting done by the system and perform respective actions.

# COMPONENT DESCRIPTION

The proposed system majorly focuses on the use of four main technologies. These technologies can be categorized as the following modules:

# DETAIL DESCRIPTION OF IVR

IVR is an advancement that allows a PC to work together with individuals utilizing voice and DTMF tones contribution through a keypad. In media interchanges, IVR licenses customers to connect with an association's host system by methods for a telephone keypad or by talk affirmation, after which organizations can be inquired about through the IVR exchange. IVR systems can respond with pre-recorded or effectively delivered sound to furthermore control customers on the most capable strategy to proceed. IVR structures sent in the framework are assessed to manage colossal call volumes and besides used for outbound calling, as IVR systems are more wise than various judicious dialer systems. The term voice reaction unit (VRU) is here and there utilized too. IVR systems can be used for convenient purchases, banking portions and organizations, retail orchestrates, utilities, travel information and atmosphere conditions Various advancements fuse using substance to-talk (TTS) to talk staggering and dynamic information, for instance, messages, news reports or atmosphere information. IVR advancement is similarly being brought into vehicle structures for without hands movement. TTS is PC created mixed talk that is never again the robotized voice generally associated with PC. Certified voices make the talk in pieces that are associated and smoothed before being played to the visitor.

The IVR proposes a few benefits that makes it an ideal technology in the development of the project.

# Increase first contact resolution:

IVR significantly increases first contact resolution because callers are always directed to the agent who is most capable of meeting their needs or the most appropriate department. The agent who receives the call will be more qualified to answer the caller’s question and will be less likely to transfer the call to another agent.

# Increase customer service efficiency:

Agents who work in a company that uses an IVR are more proficient at solving specific problems and meeting specific needs of the customers that they are assigned. The result is an increase in customer service efficiency.

# Increase agent and company efficiency:

Agents who work in a company with an IVR are more skilled at addressing specific issues, are less likely to consult with colleagues or a manager and are also less likely to

transfer the call to another agent. This results in a significant increase in agent and company efficiency.

# Reduce operational costs :

IVR systems will replace a receptionist or a customer service agent who answers calls and directs calls to agents. They are also very affordable, will increase efficiency and will reduce operational costs, so the ROI is huge.

# Increase professionalism:

You can use an IVR system to greet your customers in a very professional manner and to make it appear that you have more departments and employees than you actually have.

# Increase customer satisfaction:

When your IVR is easy to use and reliable, customers will never be routed to the wrong department, or to an agent who cannot solve their problems.

# WORKING OF SPEECH RECOGNITION

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields. Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent" systems. Systems that use training are called "speaker dependent".

Speech recognition applications include voice user interfaces such as voice dialing (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domestic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

The term voice recognition or speaker identification refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems. Speech recognition works using algorithms through acoustic and language modeling. Acoustic modeling represents the relationship between linguistic units of speech and audio signals; language modeling matches sounds with word sequences to help distinguish between words that sound similar. Often, hidden Markov models are used as well to recognize temporal patterns in speech to improve accuracy within the system. The most frequent applications of speech recognition within the enterprise include call routing, speech-to-text processing, voice dialing and voice search. While convenient, speech recognition technology still has a few issues to work through, as it is continuously developed. The pros of speech recognition software are it is easy to use and readily available. Speech recognition software is now frequently installed in computers and mobile devices, allowing for easy access. The downside of speech recognition includes its inability to capture words due to variations of pronunciation, its lack of support for most languages outside of English and its inability to sort through background noise. These factors can lead to inaccuracies.

Speech recognition performance is measured by accuracy and speed. Accuracy is measured with word error rate. WER works at the word level and identifies inaccuracies in transcription, although it cannot identify how the error occurred. Speed is measured with the real-time factor. A variety of factors can affect computer speech recognition performance, including pronunciation, accent, pitch, volume and background noise. It is important to note the terms speech recognition and voice recognition are sometimes used interchangeably. However, the two terms mean different things. Speech recognition is used to identify words in spoken language. Voice recognition is a biometric technology used to identify a particular individual's voice or for speaker identification.

# SPEECH TO TEXT CONVERTER

The process of converting spoken speech or audio into text is called speech to text converter. The process is usually called speech recognition. The Speech recognition is used to characterize the broader operation of deriving content from speech which is known as speech understanding. We often associate the process of identifying a person from their voice, which is voice recognition or speaker recognition so it is wrong to use this term for it.

To convert speech to on-screen text or a computer command, a computer has to go through several complex steps. When you speak, you create vibrations in the air. The analog-to-digital converter (ADC) translates this analog wave into digital data that the computer can understand. To do this, it samples, or digitizes, the sound by taking precise measurements of the wave at frequent intervals. The system filters the digitized sound to remove unwanted noise, and sometimes to separate it into different bands of frequency (frequency is the wavelength of the sound waves, heard by humans as differences in pitch). It also normalizes the sound, or adjusts it to a constant volume level. It may also have to be temporally aligned. People don't always speak at the same speed, so the sound must be adjusted to match the speed of the template sound samples already stored in the system's memory.

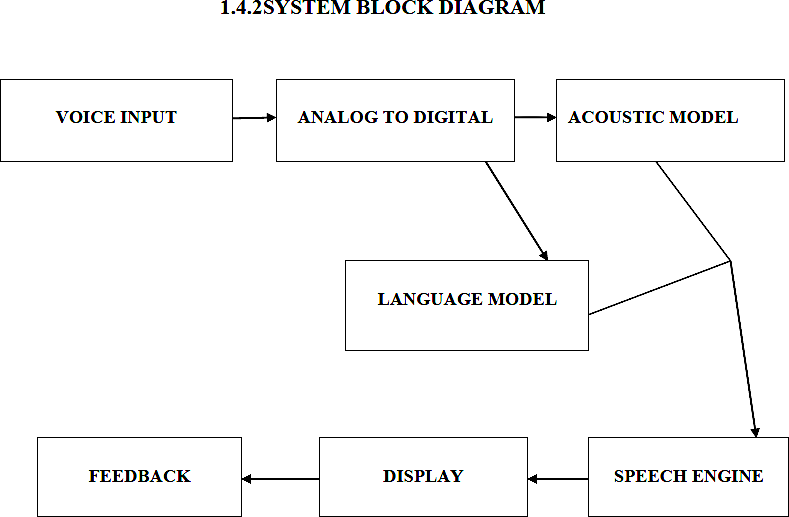


Figure-4: System Block Diagram for Speech Recognition

# SPEECH SYNTHESIS

Speech synthesis is the synthetic production of speech. An automatic data handing out system used for this purpose is called as speech synthesizer, and may be enforced in software package and hardware product. A text-to-speech (TTS) system converts language text into speech, alternative systems render symbolic linguistic representations. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text to speech program permits individual with ocular wreckage or reading disabilities to concentrate to written words on a computing device. Several computer operational systems have enclosed speech synthesizers since the first nineteen nineties’ years.

The text to speech system is consisting of 2 parts: -front-end and a back-end. The front- end consist of 2 major tasks. Firstly, its disciple unprocessed text containing symbols like numbers and abstraction into the equivalent of written out words. This method is commonly known as text, standardization, or processing. Front end then assigns spoken transcriptions to every word, and divides and marks the text into speech units, like phrases, clauses, and sentences.

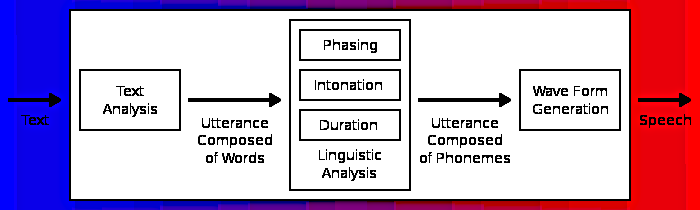
The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

Figure-5: Text to Speech Conversion

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message.

Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems. TTS is often used with voice recognition programs. There are numerous TTS products available, including Read Please 2000, Proverbe Speech Unit, and Next Up Technology's Text Aloud. Lucent, Elan, and AT&T each have products called “Text-to-Speech”.

In addition to TTS software, a number of vendors offer products involving hardware, including the Quick Link Pen from WizCom Technologies, a pen-shaped device that can scan and read words; the Road Runner from Ostrich Software, a handheld device that reads ASCII text; and Dec Talk TTS from Digital Equipment, an external hardware device that substitutes for a sound card and which includes an internal software device that works in conjunction with the PC's own sound card.

**Chapter 4**

**DESIGN**

**4.1 Design Phases of The Proposed System**

**i. Phase-1:**

The tasks that can be performed using the program developed will be prompted using the voice prompt. In background python module pyttsx3 is used for text to speech conversion.

User will be asked to provide input for the following tasks written below.

The input is expected in the form of speech by the user which will be converted to text by the Google speech application interface in python and accordingly tasks will be performed.

* **Login to their Vmail account.**
* **Send e-mail through VMail.**
* **Read e-mail through Vmail.**

**ii. Phase-2:**

In phase-2 of our program the user will give speech input to the system. This speech input will be handled by speech recognition module. It is a python library which is used to handle the voice requests and it converts speech into text. Now after receiving input from the user speech to text converter will save the response in respective variables used in the script and based on their value it will further enter into respective modules.

**iii. Phase-3:**

In this phase our program will handle the requests by the user. Based on the speech input given by the user it will launch the modules.

* Login to V-mail account:- This module will handle the request by user to login in their g-mail account. This module will make the connection with the user’s Gmail account based on the credentials provided through voice input. This module’s script designed as such it will prompt user to enter their g-mail username and password and then it will use selenium web-driver to automate the task for the user and as a result connection will be made.
* Send E-mail through V-mail:- This module will handle the request by user to send email through their g-mail account. The python script for this module will prompt the user to enter their credentials and then it will make connection with their account.

After the connection has been done it will further prompt the user to enter the receiver’s account e- mail id and it will then allow the user to speak their message and it will repeat it for them and by saying ok it will send the mail.

SMTP library in python is used for the above task.

* Read E-mail through V-mail:- This module will handle the request by user to read email through their g-mail account. The python script for this module will prompt the user to enter their credentials and then it will make connection with their account.

After the connection has been done it will start fetching the unread mails for the user and will speak it for them with the help of pyttsx3 or gTTS library in python for text to speech conversion.

**INPUT SPEECH USING RECOGNITION**

**SEND**

**MAIL**

**OUTPUT MAIL AS SPEECH USING TEXT TO SPEECH CONVERTER**

**LOGIN INTO ACCOUNT**

**CONVERT TEXT MAIL TO SPEECH**

**INPUT PASSWORD AS SPEECH**

**INPUT EMAIL ID AS SPEECH**

**TEXT**

**INPUT PASSWORD AS SPEECH**

**INPUT NAME AS SPEECH**

**INPUT EMAIL ID AS SPEECH**

**CONVERT SPEECH TO TEXT USING SPEECH TO TEXT CONVERTER**

**INPUT EMAIL ID AS SPEECH**

**REGISTER**

**LOGIN**

**READ**

**MAIL**

**INPUT PASSWORD AS SPEECH**

**LOGIN**

**COMPOSE MAIL AS SPEECH**

**SEND MAIL**

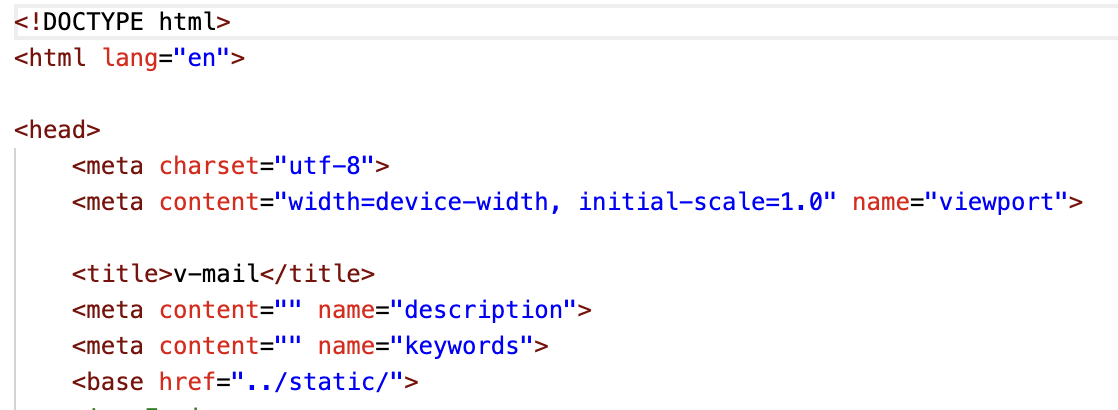
Figure-6: Illustration of Sending and Receiving E-mails

**Chapter 5**

**METHODOLOGY**

**5.1 Frontend of Voice Based Email for Blind**

* We have build this project on VS Code using HTML and CSS for frontend part.
* We are using this to make interactive interface.
* HTML is use to make basic layout of the project.

****

* CSS is used to give styling to the project.
  1. **Backend of Voice Based Email for Blind**

**5.2.1 Libraries**

* For the backend of this project we created app.py python file.
* We also installed some required libraries of python. Some of them are NumPy, python, flask, etc.

from flask import Flask,session,render\_template,request,redirect,url\_for,flash

from user import user\_operation

from encryption import Encryption

import time

from email import encoders

from gtts import gTTS

import os

from playsound import playsound

import speech\_recognition as sr

import pygame

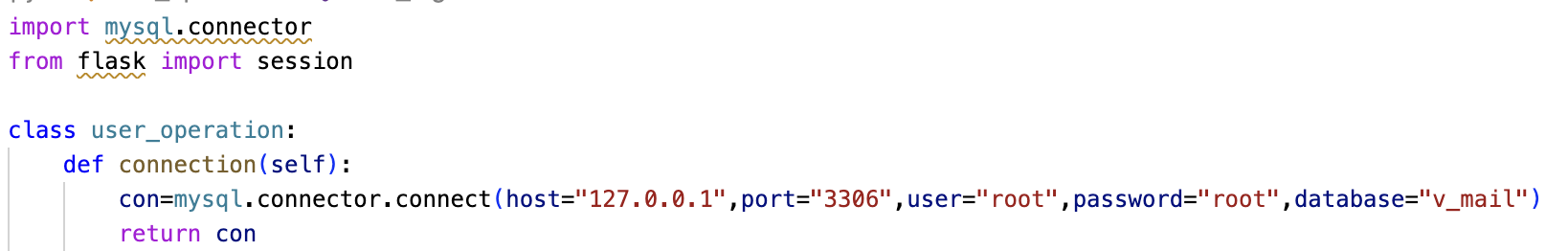
import sounddevice as sd

import soundfile as sf

import numpy

* + 1. **Databases**

**SQL (Structured Query Language)**

****

**Chapter 6**

**IMPLEMENTATION**

* 1. **SPEECH RECOGNITION IN PYTHON**

The improvement and accessibility alone in the field of speech recognition are worth considerable. It allows the physically and the elderly and visually challenged people to collaborate with state-of-the-art products and services quickly and naturally no graphical user interface is needed. If you want to use speech recognition or simply convert speech to text in your python it

is very easy to use. Let’s see how: -

* Working of speech recognition.
* Packages available in PyPI.
* How to install and how to use speech recognition package using python library.

A handful of packages for speech recognition exist on PyPI. A few of them include:

* Google-cloud-speech
* Speech Recognition

SpeechRecognition is a library that acts as a wrapper for many popular speech APIs and is thus very flexible to use. One of these is the Google Web Speech API which supports a default API key that is hard coded into the SpeechRecognition library.

The elasticity and easy to use features of the SpeechRecognition package in python make it a very good choice for developers who are working on any python project. It does not guarantee to support every feature that is wrapped with this API. You will have to dispense some time searching for the easily available options to find out if SpeechRecognition is going work in your particular case.

**6.1.1. REQUIRED INSTALLATIONS**

SpeechRecognition is the library which is compatible with Python 2.6, 2.7 and 3.3+, but it will require some additional installation steps for Python v2.0. For our project we have used Python v3.0+.

1.> pip install SpeechRecognition.

2.> pip install pyaudio.

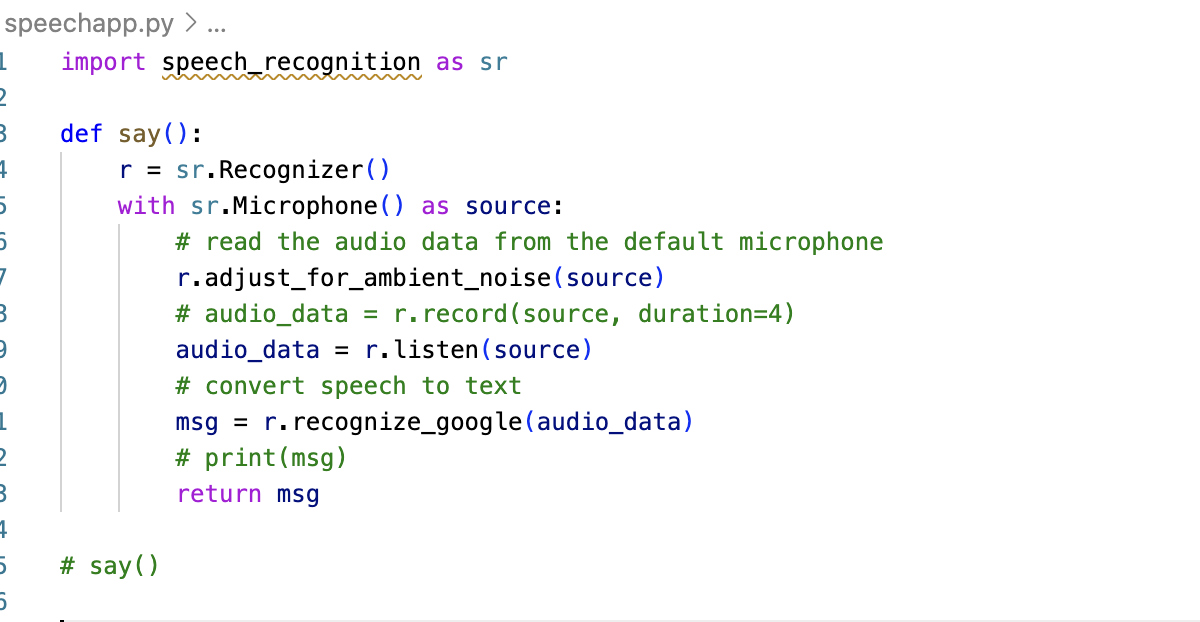
3.> pip install gtts3

4.> pip install sounddevice

5.> pip install soundfile

6.> pip install numpy

SpeechRecognition will work very good if you need to work with existing audio files. The pyaudio package comes in play when you need to capture microphone input. The main class which is used in this package is Recognizer class. The use of recognizer instance is obviously to recognize the speech. Every instance of this class comes with various settings and functionality for recognizing speech from the speaker.

**Figure-7: Speech To Text Demonstration**

The Microphone class used in this python program will let the user use the default microphone of their system instead of using some audio files as a source.

If the system of the user doesn’t have the default microphone or in case they want to use

some other microphone then they will need to specify which one to use by giving a device index. The list can be seen by calling list\_microphone\_names() which is static method of Microphone class.

Every instance of Recognizer class has seven methods for recognizing speech from speaker source using various APIs:-

* recognize\_google(): Used in “Google Web Speech API”
* recognize\_google\_cloud():Used in “Google Cloud Speech” - requires installation of the google-cloud-speech package

listen():- It is another function used for capturing microphone input. It works just like the Audio File class while Microphone is a context manager. Input can be captured from microphone using listen() method of Recognizer class. The first argument taken by this method is an audio source and it will keep on detecting the audio input until the silence is detected by it.

The audio input is generally mixed with ambient noises which can be handled by using the in-built method of recognizer class adjust\_for\_ambient\_noise().

# TEXT TO SPEECH IN PYTHON

So we have seen above what is a speech to text converter and the theory behind it but the question arises “how to implement this converter in python” so here we go.

# gTTS

Another module which can be used in python for conversion is:-

This module is Google text to speech library in python. gTTS is platform independent that is it is compatible with Windows, Linux, and MacOS speech library. This offers a great set of functionality and features.

To install this API in windows platform

**>>> pip install gTTS**



# Software Requirements:

Tools Used:

* Python IDLE.
* VS Code
* Browser
* MySQL Workbench

# Hardware Requirements:

* Windows

# Chapter 7

**TESTING AND RESULTS**

**7.1 CODE**

from flask import Flask,session,render\_template,request,redirect,url\_for,flash

from user import user\_operation

from encryption import Encryption

import time

from gtts import gTTS

import os

from playsound import playsound

import speech\_recognition as sr

import pygame

import sounddevice as sd

import soundfile as sf

import numpy

from email import encoders

file = "good"

i="0"

app=Flask(\_\_name\_\_)

app.secret\_key = 'nahsgtwjmbhkacsbkjvjvsv'

def texttospeech(text, filename):

filename = filename + '.mp3'

flag = True

while flag:

try:

tts = gTTS(text=text, lang='en', slow=False)

tts.save(filename)

flag = False

except:

print('Trying again')

# Read the audio file

data, samplerate = sf.read(filename)

# Play the sound file

sd.play(data, samplerate)

# Wait until playback is finished

sd.wait()

#playsound(filename)

os.remove(filename)

return

def speechtotext(duration):

global i, addr, passwrd

r = sr.Recognizer()

with sr.Microphone() as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

# Read the audio file

data, samplerate = sf.read('speak.mp3')

# Play the sound file

sd.play(data, samplerate)

# Wait until playback is finished

sd.wait()

#playsound('speak.mp3')

audio = r.listen(source, phrase\_time\_limit=duration)

try:

response = r.recognize\_google(audio)

except:

response = 'N'

return response

def convert\_special\_char(text):

temp=text

special\_chars = ['dotkom','dot','underscore','dollar','hash','star','plus','minus','space','dash','attherate','atedrate','at']

for character in special\_chars:

while(True):

pos=temp.find(character)

if pos == -1:

break

else :

if character == 'dotkom':

temp=temp.replace('dotkom','.com')

elif character == 'dot':

temp=temp.replace('dot','.')

elif character == 'underscore':

temp=temp.replace('underscore','\_')

elif character == 'dollar':

temp=temp.replace('dollar','$')

elif character == 'hash':

temp=temp.replace('hash','#')

elif character == 'star':

temp=temp.replace('star','\*')

elif character == 'plus':

temp=temp.replace('plus','+')

elif character == 'minus':

temp=temp.replace('minus','-')

elif character == 'space':

temp = temp.replace('space', '')

elif character == 'dash':

temp=temp.replace('dash','-')

elif character == 'attherate':

temp=temp.replace('attherate','@')

elif character == 'atedrate':

temp=temp.replace('atedrate','@')

elif character == 'at':

temp=temp.replace('at','@')

return temp

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/choice')

def choice():

global i, addr, passwrd

text1 = "Welcome to our Voice Based Email. Choose either you want to login or register give your choice "

texttospeech(text1, file + i)

i = i + str(1)

texttospeech("Enter your choice", file + i)

i = i + str(1)

choice = speechtotext(10)

if choice == 'login':

return redirect(url\_for('user\_email'))

elif(choice =='register'):

return redirect(url\_for('user\_name'))

else:

msg = "You have choose an incorrect option. please choose a valid option."

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('choice'))

@app.route('/user\_name')

def user\_name():

global i, addr, passwrd

flag = True

while flag:

texttospeech("Enter your name", file + i)

i = i + str(1)

name = speechtotext(10)

if name != 'N':

texttospeech("You meant " + name + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(3)

if say.lower() == 'confirm' or say.lower() == 'yes':

flag = False

return render\_template('user\_name.html',name=name)

else:

texttospeech("Could not understand what you meant:", file + i)

i = i + str(1)

return redirect(url\_for('user\_name'))

@app.route('/user\_create\_email')

def user\_create\_email():

global i, addr, passwrd

flag = True

while flag:

texttospeech("create your Email id", file + i)

i = i + str(1)

name=request.args.get('name')

email = speechtotext(10)

if email != 'N':

texttospeech("You meant " + email + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(3)

if say.lower() == 'confirm' or say.lower() == 'yes':

email = email.strip().replace(' ', '').lower()

email = convert\_special\_char(email)

flag = False

return render\_template('user\_create\_email.html',email=email,name=name)

else:

texttospeech("Could not understand what you meant:", file + i)

i = i + str(1)

return redirect(url\_for('user\_create\_email',name=name))

@app.route('/user\_create\_password',methods=['GET','POST'])

def user\_create\_password():

global i, addr, passwrd

flag = True

while flag:

texttospeech("create your email password", file + i)

i = i + str(1)

email=request.args.get('email')

name=request.args.get('name')

password = speechtotext(10)

if password != 'N':

texttospeech("You meant " + password + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(3)

if say.lower() == 'confirm' or say.lower() == 'yes':

password = password.strip().replace(' ', '').lower()

password = convert\_special\_char(password)

flag = False

return render\_template('user\_create\_password.html',password=password,email=email,name=name)

else:

texttospeech("Could not understand what you meant:", file + i)

i = i + str(1)

return redirect(url\_for('user\_create\_password',email=email,name=name))

@app.route('/user\_signup',methods=['GET','POST'])

def user\_signup():

global i, addr, passwrd

if(request.method=='POST'):

name = request.form['name']

email = request.form['email']

password = request.form['password']

e = Encryption()

password = e.convert(password)

ob = user\_operation()

ob.user\_signup(name,email,password)

msg = "Successfully Registered. Thank you. You can login now."

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_email'))

@app.route('/user\_email')

def user\_email():

global i, addr, passwrd

flag = True

while flag:

texttospeech("Enter your Email", file + i)

i = i + str(1)

addr = speechtotext(10)

if addr != 'N':

addr = addr.strip().replace(' ', '').lower()

addr = convert\_special\_char(addr)

flag = False

return render\_template('user\_email.html',email=addr)

else:

texttospeech("Could not understand what you meant:", file + i)

i = i + str(1)

@app.route('/confirm\_user\_email')

def confirm\_user\_email():

global i

email = request.args.get('email')

texttospeech("You meant " + email + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(3)

if say.lower() == 'confirm' or say.lower() == 'yes':

return redirect(url\_for('user\_password'))

else:

return redirect(url\_for('user\_email'))

@app.route('/user\_password',methods=['GET','POST'])

def user\_password():

global i, addr, passwrd

flag = True

while flag:

texttospeech("Enter your password", file + i)

i = i + str(1)

passwrd = speechtotext(10)

if passwrd != 'N':

texttospeech("You meant " + passwrd + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(3)

if say.lower() == 'confirm' or say.lower() == 'yes':

passwrd = passwrd.strip().replace(' ', '').lower()

passwrd = convert\_special\_char(passwrd)

flag = False

return render\_template('user\_password.html',password=passwrd,email=addr)

else:

texttospeech("Could not understand what you meant:", file + i)

i = i + str(1)

@app.route('/user\_login\_verify',methods=['GET','POST'])

def user\_login\_verify():

global i, addr, passwrd

if(request.method=='POST'):

email = request.form['email']

password = request.form['password']

e = Encryption()

password = e.convert(password)

ob = user\_operation()

rc = ob.user\_login(email,password)

if(rc==0):

msg = "Invalid email Or password please try again"

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_email'))

else:

msg = "Welcome "+ session['name']

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_dashboard'))

@app.route('/user\_logout')

def user\_logout():

global i, addr, passwrd

if('email' in session):

session.clear()

msg = "Logged out successfully"

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_login'))

else:

return redirect(url\_for('index'))

@app.route('/user\_dashboard')

def user\_dashboard():

global i, addr, passwrd

flag = True

while flag:

if 'email' in session:

msg = "Welcome " + session['name'] + " to V Mail"

texttospeech(msg, file + i)

i = i + str(1)

ob = user\_operation()

record = ob.user\_inbox\_mails()

return render\_template('user\_dashboard.html', record=record)

else:

msg = "You are not authorized! Please login first"

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_email'))

@app.route('/user\_command')

def user\_command():

global i, addr, passwrd

flag = True

while flag:

try:

if('email' in session):

msg = "Please say OK MAIL to get your virtual assistant"

texttospeech(msg, file + i)

i = i + str(1)

say = speechtotext(8)

if say.lower() == "ok mail":

flag = False

return redirect(url\_for('user\_assistant'))

else:

texttospeech("invalid command", file + i)

i = i + str(1)

except Exception as e:

msg = "Your Voice is not audible."

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_command'))

@app.route('/user\_assistant')

def user\_assistant():

global i, addr, passwrd

flag = True

while flag:

if('email' in session):

msg = "Please tell a command to proceed like compose mail, read mail or logout"

texttospeech(msg, file + i)

i = i + str(1)

command = speechtotext(5)

if(command == "compose mail"):

return redirect(url\_for('user\_compose\_to'))

elif(command == "read mail"):

return redirect(url\_for('user\_read\_mail'))

elif(command == "logout" ):

return redirect(url\_for('user\_logout'))

else:

msg = "invalid command"

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_email'))

@app.route('/user\_compose\_to')

def user\_compose\_to():

global i, addr, passwrd, s, item, subject, body

text1 = "You have reached the page where you can compose and send an email. "

texttospeech(text1, file + str(i))

i = str(int(i) + 1)

flag = True

flag1 = True

fromaddr = addr

toaddr = list()

while flag1:

while flag:

texttospeech("Enter receiver's email address:", file + str(i))

i = str(int(i) + 1)

to = speechtotext(15)

if to != 'N':

texttospeech("You meant " + to + ". please 'confirm' or say 'no' to enter again", file + str(i))

i = str(int(i) + 1)

say = speechtotext(5)

if say.lower() == 'confirm' or say.lower() == 'yes':

to = to.strip().replace(' ', '').lower()

to = convert\_special\_char(to)

toaddr.append(to)

flag = False

return render\_template('user\_compose\_to.html', receiver=to)

else:

texttospeech("Could not understand what you meant", file + str(i))

i = str(int(i) + 1)

return redirect(url\_for('user\_compose\_to'))

texttospeech("Do you want to enter more recipients? Say 'yes' or 'no'.", file + str(i))

i = str(int(i) + 1)

say1 = speechtotext(3)

if say1.lower() == 'no':

flag1 = False

flag = True

newtoaddr = list()

for item in toaddr:

item = item.strip()

item = item.replace(' ', '')

item = item.lower()

item = convert\_special\_char(item)

newtoaddr.append(item)

print(item)

@app.route('/user\_compose\_subject')

def user\_compose\_subject():

global i, addr, passwrd, s, item, subject, body

flag = True

while (flag):

texttospeech("enter subject", file + i)

i = i + str(1)

rec = request.args.get('rec')

subject = speechtotext(10)

if subject != 'N':

texttospeech("You meant " + subject + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(5)

if say.lower() == 'confirm' or say.lower() == 'yes':

flag = False

return render\_template('user\_compose\_subject.html',receiver=rec,subject=subject)

else:

texttospeech("could not understand what you meant", file + i)

i = i + str(1)

return redirect(url\_for('user\_compose\_to',receiver = rec))

@app.route('/user\_compose\_message')

def user\_compose\_message():

global i, addr, passwrd, s, item, subject, body

flag = True

while flag:

texttospeech("enter body of the mail", file + i)

i = i + str(1)

rec = request.args.get('rec')

sub = request.args.get('sub')

body = speechtotext(20)

if body != 'N':

texttospeech("You meant " + body + ". please 'confirm' or say 'no' to enter again", file + i)

i = i + str(1)

say = speechtotext(5)

if say.lower() == 'confirm' or say.lower() == 'yes':

flag = False

return render\_template('user\_compose\_message.html',receiver=rec,subject=sub,message=body)

else:

texttospeech("could not understand what you meant", file + i)

i = i + str(1)

return redirect(url\_for('user\_compose\_to',receiver = rec,subject = sub))

@app.route('/user\_compose\_mail', methods=['GET', 'POST'])

def user\_compose\_mail():

global i, addr, passwrd

if 'email' in session:

msg = "Are you sure you want to 'send' this mail or want to 'cancel'?"

texttospeech(msg, file + i)

i = i + str(1)

if request.method == 'POST':

rec = request.form['receiver']

sub = request.form['subject']

message = request.form['message']

else:

rec = request.args.get('receiver')

sub = request.args.get('subject')

message = request.args.get('message')

try:

data = speechtotext(5)

if data:

if data.lower() == 'yes' or data.lower() == 'send':

ob = user\_operation()

ob.user\_send\_mail(rec, sub, message)

msg = "Mail sent successfully"

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_dashboard'))

elif data.lower() == 'cancel':

msg = "Your message is not sent"

texttospeech(msg, file + i)

i = i + str(1)

return redirect(url\_for('user\_dashboard'))

else:

msg = "Invalid command"

texttospeech(msg, file + str(i))

i += 1

return redirect(url\_for('user\_compose\_mail', receiver=rec, subject=sub, message=message))

except Exception as e:

msg = "Your voice is not audible."

texttospeech(msg, file + str(i))

i = i + str(1)

return redirect(url\_for('user\_compose\_mail', receiver=rec, subject=sub, message=message))

else:

msg = "You are not logged in yet"

texttospeech(msg, file + i)

i = i + str(1)

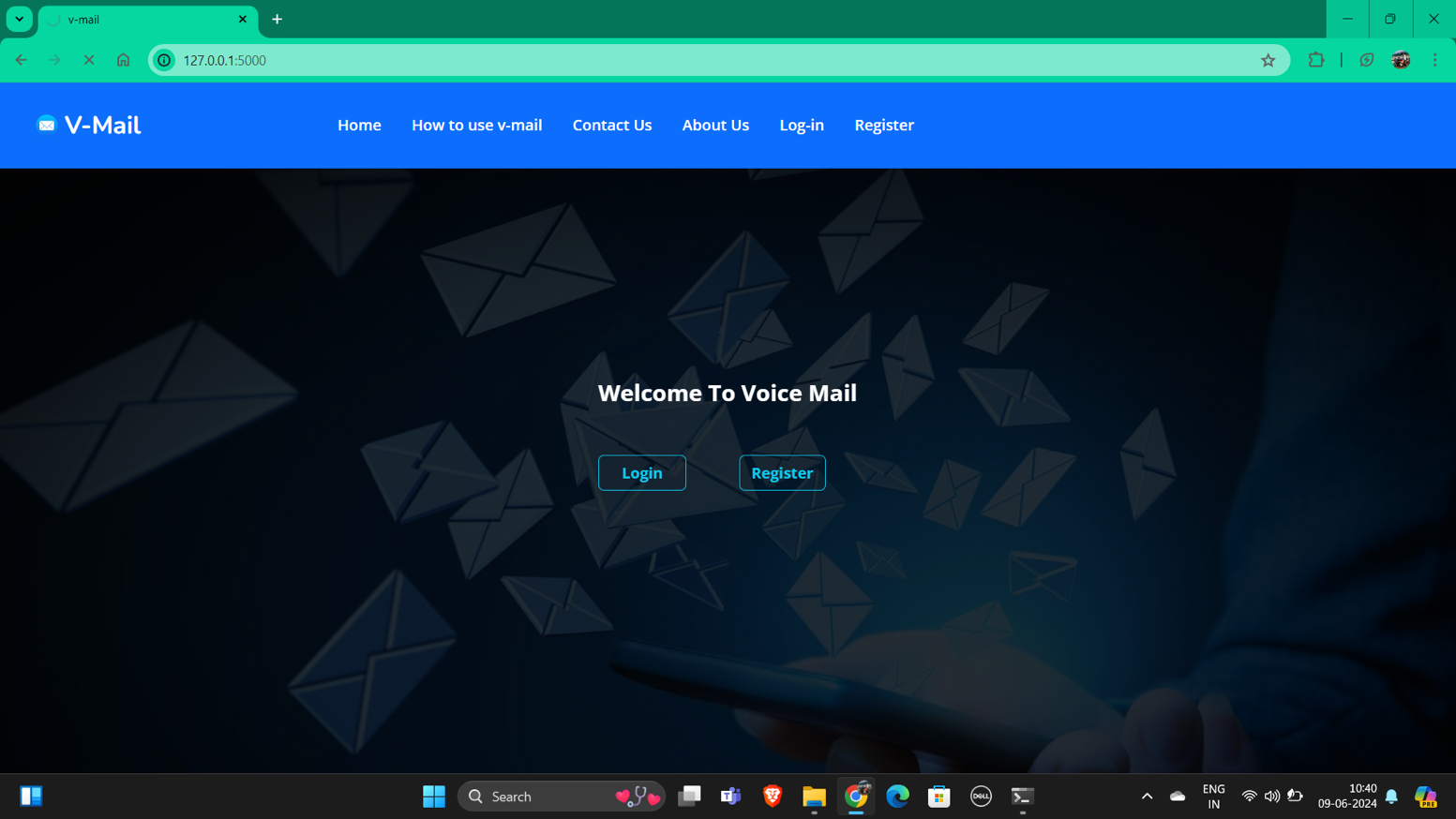
return redirect(url\_for('user\_email'))

if \_\_name\_\_==("\_\_main\_\_"):

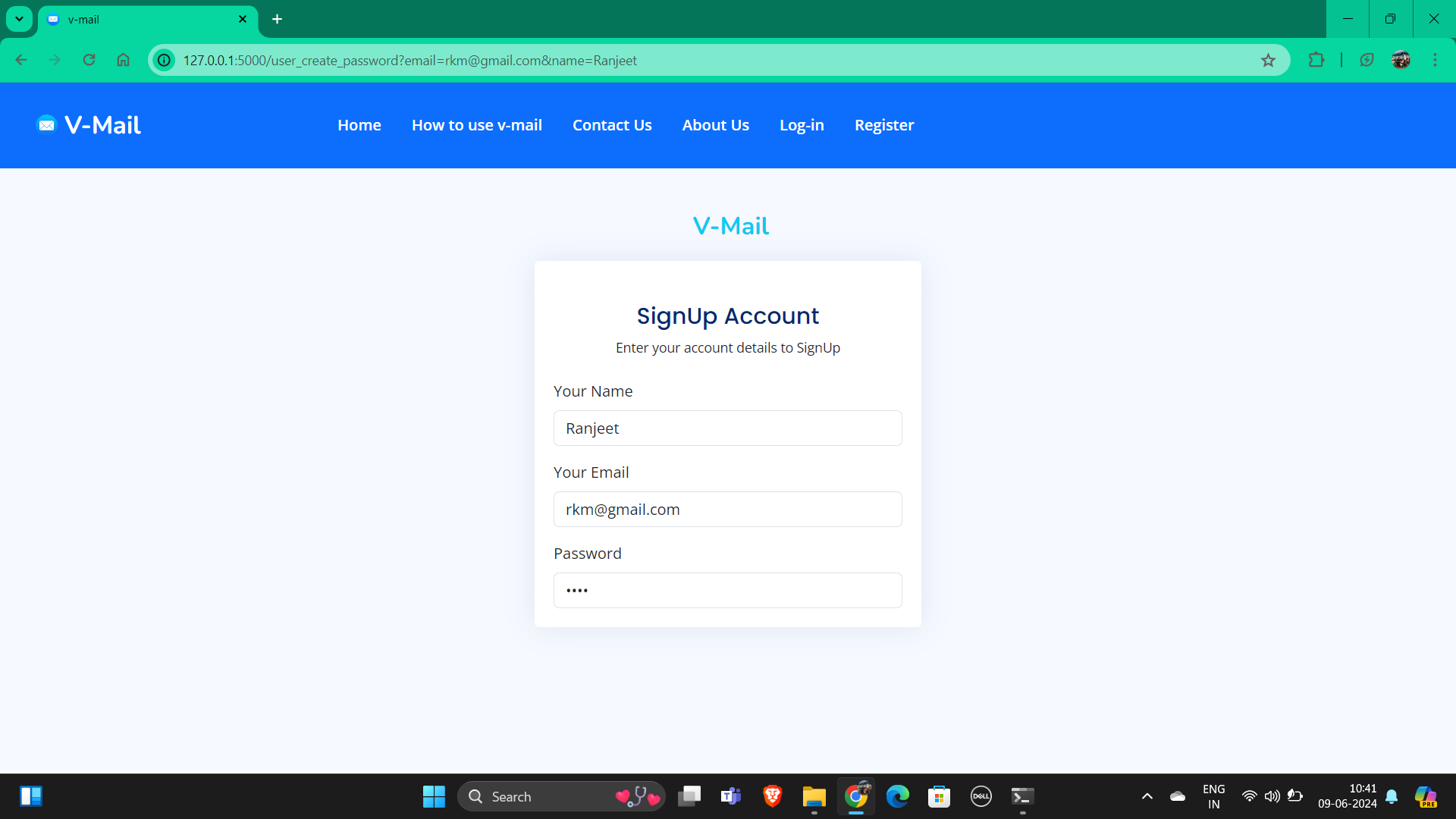
app.run(debug=True)

**7.2. Screenshot**

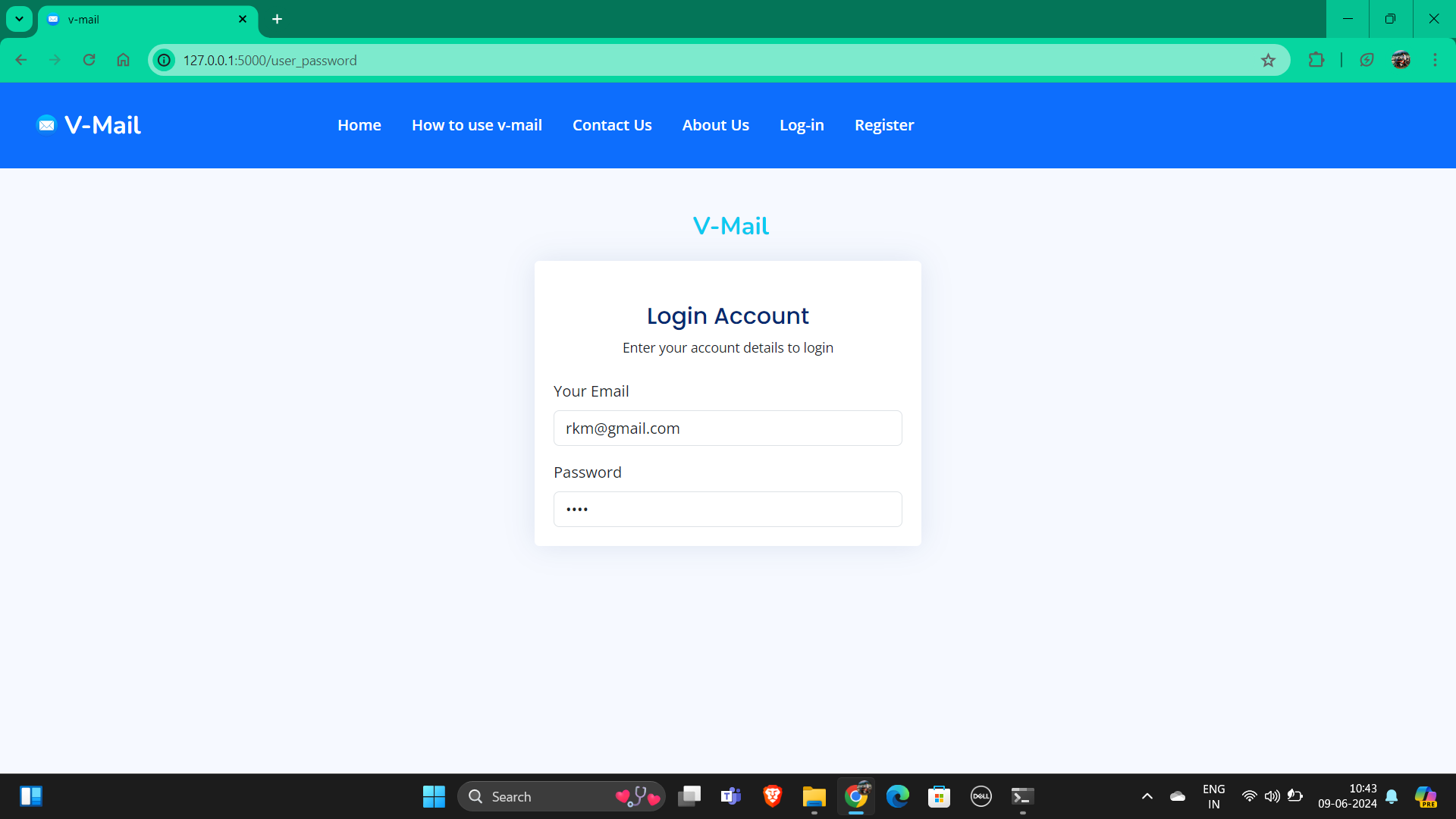
**1. Home Page**

****

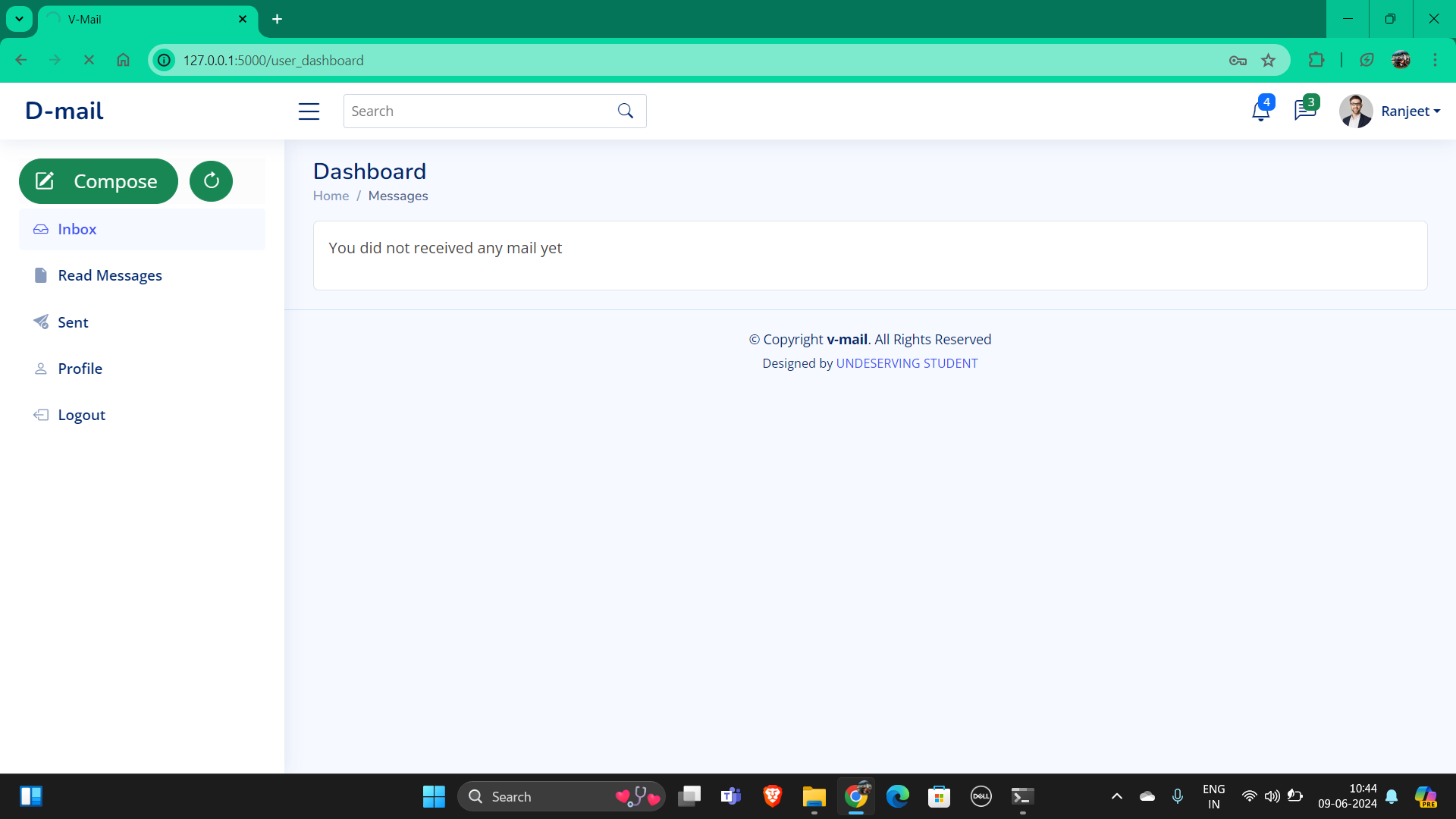
**2. Register Page**

****

**3. Login Page**

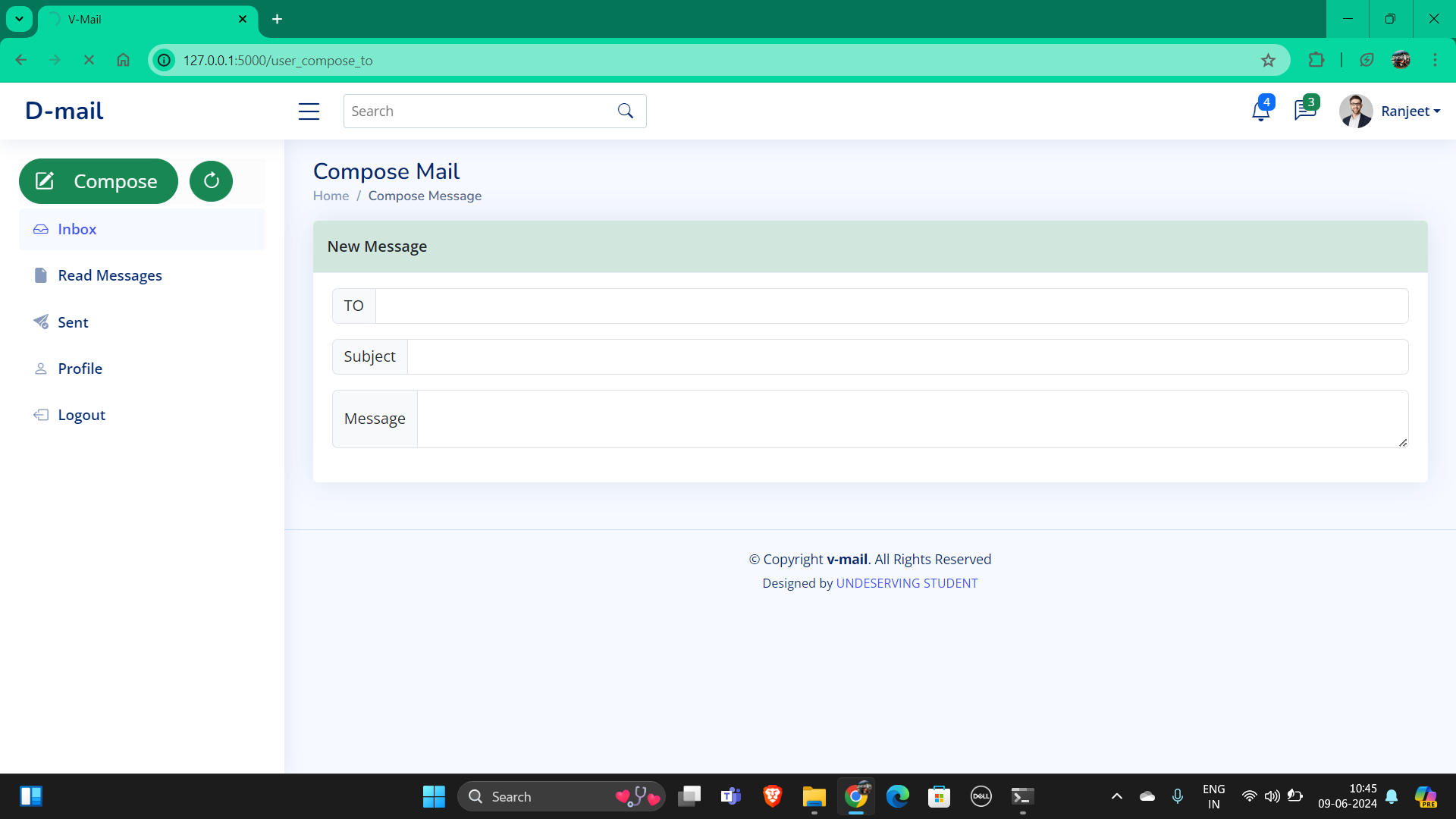
****

**4. Dashboard Page**

****



**5. Compose Mail**

****



**Chapter 8**

**CONCLUSION**

The project of voice-based email system for blind's major goal is to develop email communication for blind individuals using voice commands due to their incapacity to utilize the internet and its various functionalities. We were successful in receiving unseen messages and providing the sender's mail Id, topic, and message.

This e-mail system can be used by any user of any age group with ease of access. It has highlight of speech to content just as content to speech with discourse reader which makes planned framework to be taken care of by outwardly hindered individual too. Now the visually impaired people can send and receive mails with a lot of ease only through voice commands without making any use of a keyboard or any mouse. It has helped in eradicating the difficulties that the blind people face and them more in the form of a normal individuals.

It has wiped out the idea of utilizing console easy routes alongside screen readers which will help decreasing the intellectual heap of recollecting console alternate ways. Also, any non-sophisticated user who does not know the position of keys on the keyboard should not bother as keyboard usage is eliminated. Instructions given by the IVR accordingly to get the respective services offered.

# FUTURE SCOPE

It is an observation that about 70% of total blind population across the world is present in INDIA. This depicts the voice message engineering utilized by daze individuals to get to E-mail and multimedia elements of working framework effectively and efficiently. Separated from this, the uneducated, crippled and daze individuals will too be able to send in their local dialects. This design will likewise decrease intellectual burden taken by blinds to recall and type characters utilizing console. Advances in technology will allow consumers and business to implement speech recognition systems at a relatively low cost and efficiently. Apart from this the system can be enhanced to help the illiterate people by making speech recognition possible in their native languages

# ADVANTAGES

* The disabilities of visually impaired folks are thrashed.
* This method makes the disabled folks desire a standard user.
* Completely voice based, wiped out the use of keyboard and mouse.
* Efficient and robust
* This design also scales back psychological feature load taken by blind to recollect and kind characters mistreatment keyboard.
* User friendly

# REFERENCES

1. Sherly Noel, Human computer interaction based Smart Voice Email Application - Assistant for Visually Impaired Users, IEEE Xplore International Conference on Smart Systems and Inventive Technology, 2020, ICSSIT.
2. Paulus A. Tiwari, Pratiksha Zodawan, Harsha P. Nimkar, Trishna Rotke, Priya G. Wanjari, Umesh Samarth, A Review on Voice based E-Mail System for Blind, IEEE Xplore International Conference on Inventive Computation Technologies, 2020, ICICT.
3. Sherly Noel, Email System for Visually Impaired People, 2021 International Research Journal of Engineering and Technology (IRJET), Vol.07, Issue.03.
4. Aman Pratap Kushwaha, Aayushmaan, Harendra Singh, Voice-Based Email System, Vol. 6, Issue. 7, July – 2021 IJISRT.
5. Rahul Kumar, Vaishali Singh, Dr. Nikhat Akhtar, MrsVersha ‬ Verma,