**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belagavi-590018**





A Project Report on

**“VOICE BASED EMAIL SYSTEM”**

A project work submitted in partial fulfilment of the requirement for the award of the degree of

**Bachelor in Engineering**

**In**

**Computer Science & Engineering**

**Submitted By**

**USHA K (3SL17CS023)**

**SANIYAH RIFATH (3SL17CS406)**

**N.BHAVYADEVI VISHALAKSHI (3SL17CS012)**

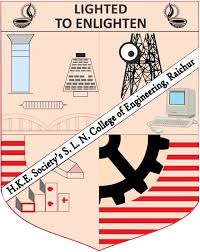
**NAGA NEHA LALITHA (3SL17CS013)**

**Under the guidance of**

**Prof. SUMANGALA.I**

**H.O.D**

**Department of Computer Science & Engineering**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**H.K.E Society’s S.L.N. COLLEGE OF ENGINEERING** (Affiliated to Visvesvaraya Technological University, Belgaum) Yermaras Camp, Raichur – 584135.

**2020-2021.**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belagavi-590018**

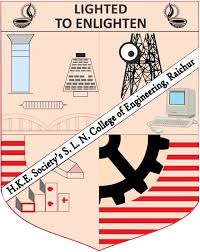




**H.K.E SOCIETY’S**

**S.L.N.COLLEGE OF ENGINEERING**

RAICHUR-584135



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

Certified that the project entitled **“VOICE BASED EMAIL SYSTEM”** carried out by a bonafied students of **B.E. 8th semester** in **COMPUTER SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM** during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the Departmental library. The Project Report has been approved as it satisfies the academic requirements in respect of project prescribed for the said Degree.

|  |  |  |
| --- | --- | --- |
| ***Signature of Guide*** | ***Signature of Coordinator*** | ***Signature of HOD*** |
| **Prof. SUMANGALA.I** | **Prof. GEETHA .B** | **Prof. SUMANGALA.I** |

Name of the Student University Seat Number

Usha k (3sl17cs023)

Saniyah Rifath (3sl18cs406)

N Bhavyadevi Vishalakshi (3sl17cs012)

Naga Neha Lalitha (3sl17cs013)

Name of the examiners Signature with Date

1.

2.

**ACKNOWLEDGEMENT**

I consider a privilege to express my deep sense of gratitude and respect to all those, who guided and inspired for the successful completion of this Internship.

My sincere thanks to our beloved Principal **Dr. R. BASAVARAJA**, H.K.E Society’s S.L.N. College Of Engineering, Raichur for his kind Co-operation during our entire course.

I consider it a privilege to express my thanks to our beloved HOD, **Prof. SUMANGALA. I,**

Department of Computer Science & Engineering, for valuable support and Guidance.

My sincere thanks to our beloved Internship coordinator **Prof. GEETHA. B,** Department of Computer Science & Engineering, for her kind support and cooperation.

My sincere thanks to our beloved Internship Guide **Prof. SUMANGALA. I,** Department of Computer Science & Engineering, for her kind support and cooperation.

My sincere thanks to all the faculty of department of Computer Science & Engineering, for their kind support and Cooperation.

I am thankful to my parents and my friends for their encouragement, affection and moral support showered on me during my entire course.

### Usha k (3sl17cs023)

### Saniyah Rifath (3sl18cs406)

### N Bhavyadevi Vishalakshi (3sl17cs012)

### Naga Neha Lalitha (3sl17cs013)

**ABSTRACT**

Internet has become one of the basic amenities for day-to-day living. Every human being is widely accessing the knowledge and information through internet. However, blind people face difficulties in accessing these text materials, also in using any service provided through internet. The advancement in computer based accessible systems has opened up many avenues for the visually impaired across the globe in a wide way. Audio feedback based virtual environment like, the screen readers have helped Blind people to access internet applications immensely. 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 with the help of a computer.

This project introduces the Voicemail system structural design that can be used by a blind person to access E-Mails easily. The involvement of research is helping blind individual to send and receive voice-based mails messages in English language with the help of a computer.

**CONTENTS**

**SL. NO. TITLE PAGE NO.**

01 INTRODUCTION 06

02 OBJECTIVES 07

03 LITERATURE REVIEW 08

04 SYSTEM REQUIREMENTSSPECIFICATIONS 10

* SOFTWARE REQUIREMENTS
* HARDWARE REQUIREMENTS
* SPEECH RECOGNITION
* SPEECH TO TEXT CONVERTER(STT)
* TEXT TO SPEECH CONVERTER(TTS)

1. DESIGN METHODOLOGY 16

* DFD
* USE CASE DAIGRAM
* SYSTEM ARCHITECTURE

06 SYSTEM IMPLEMENTATION 19

* EXISTING SYSTEM
* PROPOSED SYSTEM
* ADVANTAGES
* APPLICATIONS

07 SOURCECODE AND SCREENSHOTS 21

08 CONCLUSION 28

09 FUTURE SCOPE 29

10 REFERENCES 30

1. **INTRODUCTION**

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, 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 not 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, he/she wants to achieve.

1. **OBJECTIVES**

This project proposes a python-based application, designed specifically for visually impaired people. This application provides a voice-based mailing service where they could read and send mail on their own, without any guidance through their g-mail accounts. Here, the users have to use certain keywords which will perform certain actions for e.g. Read Mails, Compose Mail etc. 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.

1. **LITERATURE REVIEW**

**“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 cues as to the state of a transaction, and Mark-up 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.

1. **SYSTEM REQUIREMENT SPECIFICATIONS**

**HARDWARE REQUIREMENTS:**

* **Computer processor:** Pentium 1.1 GHz.
* **Hard Disk Space:** 20 GB and more.
* **RAM:** 512 MB.

**SOFTWARE REQUIREMENTS:**

* Operating system: Windows 7 and more
* Programming language**:** PYTHON
* Front-end Framework: DJANGO
* Front-end technology used: HTML,CSS,JAVASCRIPT,JQUERY,AJAX

**Python:**

Python is used for creating backbone structure. Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, it uses whitespace indentation, rather than curly braces or keywords. Python has a large standard library, commonly cited as one of Python's greatest strengths.

**Interactive Voice Response (IVR):**

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. IVR systems deployed in the network are sized to handle large call volumes and also used for outbound calling, as IVR systems are more intelligent than many predictive dialer systems.

IVR systems can be used for mobile purchases, banking payments and services, retail orders, utilities, travel information and weather conditions. A common misconception refers to an automated attendant as an IVR. The terms are distinct and mean different things to traditional telecommunications professionals—the purpose of an IVR is to take input, process it, and return a result, whereas that of an automated attendant is to route calls. The term voice response unit (VRU) is sometimes used as well. DTMF decoding and speech recognition are used to interpret the caller's response to voice prompts. DTMF tones are entered via the telephone keypad.

Other technologies include using text-to-speech (TTS) to speak complex and dynamic information, such as e-mails, news reports or weather information. IVR technology is also being introduced into automobile systems for hands-free operation. TTS is computer generated synthesized speech that is no longer the robotic voice traditionally associated with computers. Real voices create the speech in fragments that are spliced together (concatenated) and smoothed before being played to the caller.

Another technology which can be used is using text to speech to talk advanced and dynamic data, such as e-mails, reports and news and data about weather. IVR used in automobile systems for easy operations too. Text to Speech is system originated synthesized speech that’s not the robotic voice historically related to computer. Original voices produce the speech in portions that are joined together and rounded before played to the caller.

**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 "enrolment") where an individual speaker reads text or isolated vocabulary into the system. The system analyses 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 dialling (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), demotic 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 modelling. Acoustic modelling represents the relationship between linguistic units of speech and audio signals; language modelling 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 dialling 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 (STT):**

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, that is voice recognition or speaker recognition so it is wrong to use this term for it.

Systems generally use the pronunciation model. It is really imperative to learn that there is nothing like a universal speech recognizer. If you want to get the best quality of transcription, you can specialize the above models for the any given language communication channel.

Likewise, another pattern recognition technology, speech recognition can also not be without error. Accuracy of speech transcript deeply relies on the voice of the speaker, the characteristic of speech and the environmental conditions. Speech recognition is a tougher method than what folks unremarkably assume, for a personality’s being. Humans are born for understanding speech, not to transcribing it, and solely speech that’s well developed will be transcribed unequivocally. From the user's purpose of read, a speech to text system will be categorised based in its use.

**Text to Speech Converter (TTS):**

Speech synthesis is the synthetic production of speech. A 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.





Linguistic

Analysis

Wave Form

Generation

Text

Analysis

Text

Utterance

Composed

Of Phonemes

Utterance

Composed

Of words

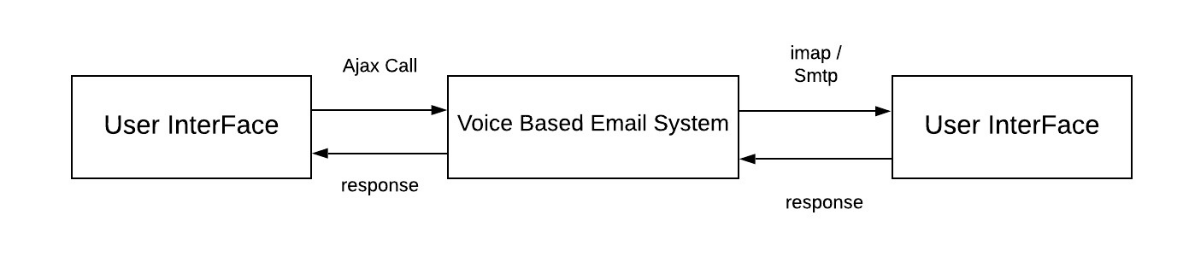
Speech

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, Proverb 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 DecTalk 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.

1. **DESIGN METHODOLOGY**

**DATA FLOW DIAGRAM:**

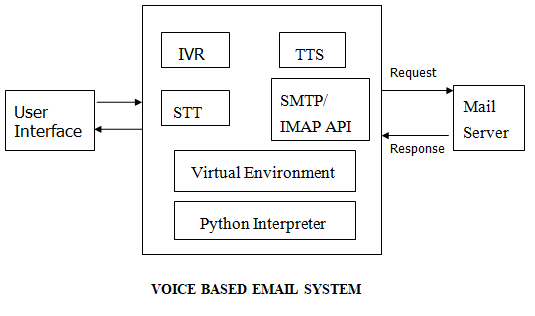
**Level1**

**USE CASE DIAGRAM :**

Diagram

Description automatically generated

**SYSTEM ARCHITECTURE:**

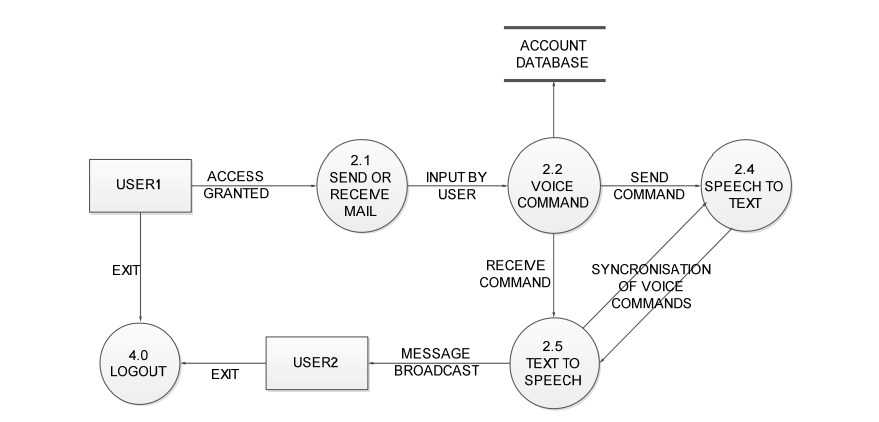
****

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 who’s 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?
* 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. The system is currently developed by us. When user will visit our site he would first have to register in our website through registration form. User will be very well guided with the help of voice commands, while registration all the necessary fields to be filled will be read by site, by clicking on that box he would have to fill in them. For e.g. If cursor moves over register icon it would sound “register button” , after clicking on register button it would sound like “you are on registration page”.



While filling up the necessary fields, speech would be recorded in database. Very frequently used words will be present i.e., when user would speak it would get typed automatically.

Also, the voice would be recorded in the database. Because after registration, user has to go to login page and type user id & password which would get recognized through database enabling the correct user to get access to his/her account. After successful login the user would read the received mails present in inbox and also can send the mails.

1. **SYSTEM IMPLEMENTATION**

**EXISTING SYSTEM:**

Simple e-mail systems are available in which only voice recognition & text-to-speech systems are accessible. The voice based e-mail system proposed by T.Shabana, A.Anam, A.Rafiya, K.Aisha has made use of IVR, Speech to text converter, Mouse click event and Screen reader. Input is based on speech & mouse clicks to give output.

The most common mail services that we use in our day to day life cannot be used by visually challenged people. This is because they do not provide any facility so that the person in front can hear out the content of the screen. As they cannot visualize what is already present on screen they cannot make out where to click in order to perform the required operations. Although there are many screen readers available then also these people face some minor difficulties. Screen readers read out whatever content is there on the screen and to perform those actions the person will have to use keyboard shortcuts as mouse location cannot be traced by the screen readers. A user is new to computer can therefore not use this service as they are not aware of the key locations.

**PROPOSED SYSTEM:**

The planned system is relies on a very fresh plan and obscurity just like the accessible mail systems. The foremost necessary facet that erstwhile unbroken in brain whereas developing the planned system’s accessibility.

The present systems don’t give this much convenience. So the systems present have a tendency to area unit developing is totally dissent from this system. In contrast to present system which emphasize more on user easiness of naive users, this system focus more on user easiness of all kind of folks including naive folks visually disabled people as well as uneducated people.

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.

The entire structure is based on IVR- interactive voice response. When using this system the computer will prompt the client to perform precise operations to gain relevant services and if the client needs to way in the relevant services then they need to perform that particular operation. One of the most important recompense of this system is that user will not need to use the keyboard. All operations will be based on voice proceedings.

**ADVANTAGES:**

* The disabilities of visually impaired people are Thrashed.
* This system makes the disabled people feel like a normal user.
* They can hear the recently received mails to the Inbox, as well as the IVR technology proves very effective for them in the terms of guidance.
* The visually impaired people can advance from Desktop application to the web-based application.

**APPLICATIONS:**

* It is basically designed to help handicapped people, who face difficulties in accessing the computer system.
* It is used to reduce cognitive load taken by blind to remember and type characters using keyboard.

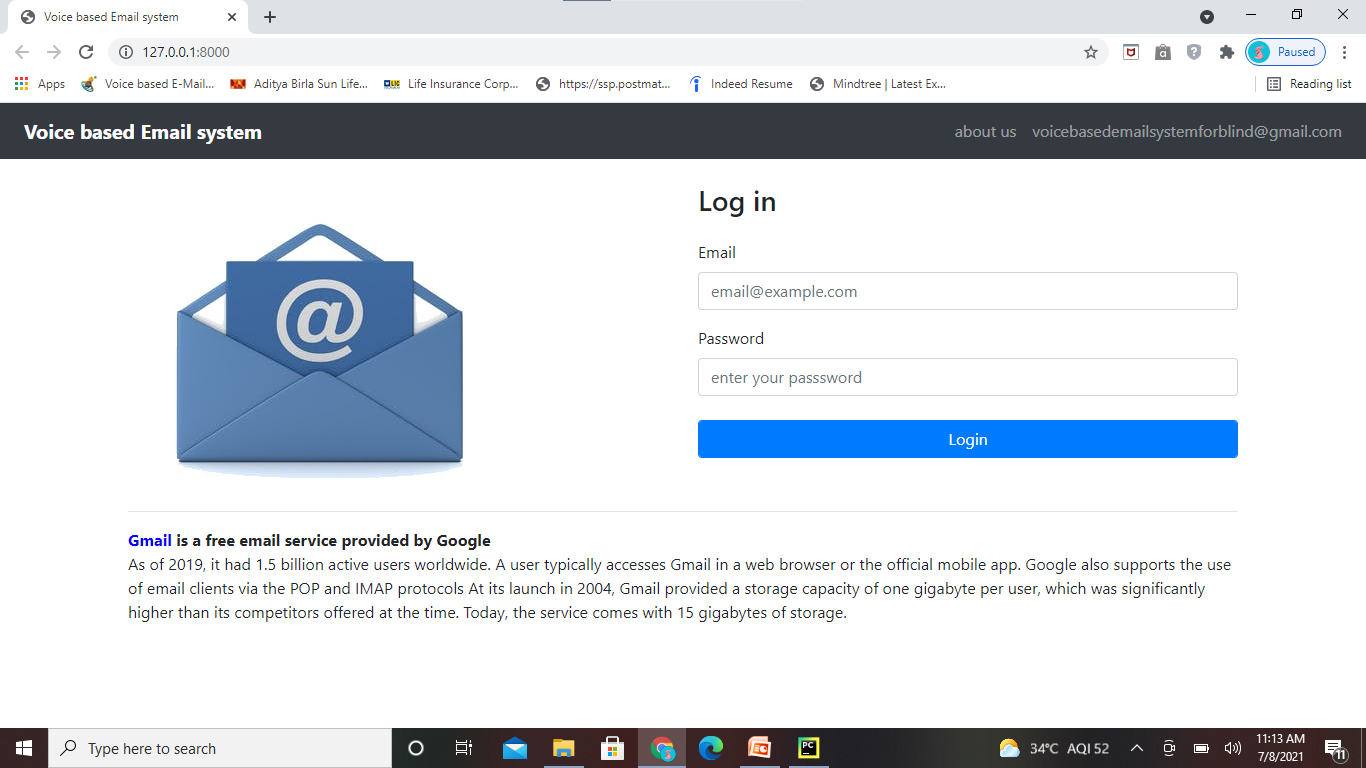
This project is proposed for the betterment of society. This project aims to help the visually impaired people to be a part of growing digital India by using internet and also aims to make life of such people quite easy. Also, the success of this project will also encourage developers to build something more useful for visually impaired or illiterate people, who also deserve an equal standard in society.

1. **SOURCECODE AND SCREENSHOTS.**

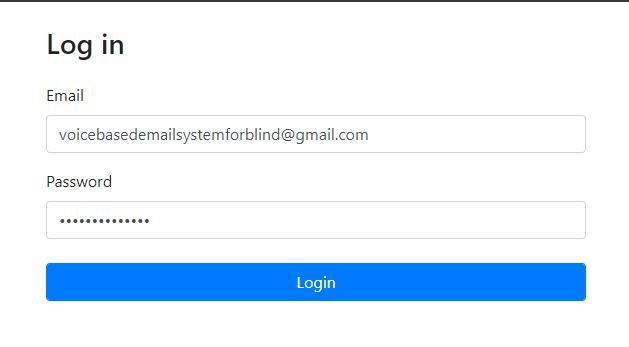
**SOURCECODE: from** django.shortcuts **import** render, redirect  
**import** pandas **as** pd  
**from** django.http **import** HttpResponse  
**from** .utils **import** sayToUserTTS, getUserInputSTT,readMailsContext  
**import** json  
**import** smtplib, ssl  
**import** imaplib  
**import** email  
**import** email.mime.text  
**import** time  
**import** threading  
name = **'vmail'  
  
def** welcome(request):  
 sayToUserTTS(**"welcome to Voice based email system"**)  
 **return** HttpResponse()  
  
**def** emailId(request):  
 emailid = getUserInputSTT(**"say your email id"**,7)  
 emailid = emailid.replace(**' '**, **''**).lower().replace(**'dot'**, **'.'**).replace(**'attherate'**, **'@'**)  
 **return** HttpResponse(emailid)  
  
**def** password(request):  
 password = getUserInputSTT(**"say your password"**)  
 password = password.replace(**' '**, **''**).lower().replace(**'dot'**, **'.'**).replace(**'attherate'**, **'@'**)  
 **return** HttpResponse(password)  
  
**def** getCommand(request):  
 sayToUserTTS(**"Hi i am "** + name + **" i can send and read your mails"**)  
 command = getUserInputSTT(**"how can i help you"**)  
  
 **if 'send mail' in** command.lower():  
 **return** HttpResponse(json.dumps({**'status'**: 200, **'url'**: **'sendmail'**}))  
 **elif 'read mail' in** command.lower():  
 **return** HttpResponse(json.dumps({**'status'**: 200, **'url'**: **'readmails'**}))  
 **elif 'logout' in** command.lower() **or 'log out' in** command.lower():  
 **del** request.session[**'email\_id'**]  
 **del** request.session[**'password'**]  
 **return** HttpResponse(json.dumps({**'status'**: 400, **'url'**: **'login'**}))  
 **else**:  
 sayToUserTTS(**"sorry i did not understand your command"**)  
 **return** HttpResponse(json.dumps({**'status'**: 400, **'url'**: **'login'**}))  
  
**def** login(request):  
 **if** request.method == **'POST'**:  
 emailid = request.POST.get(**'emailid'**)  
 password = request.POST.get(**'password'**)  
 **try**:  
 mail = imaplib.IMAP4\_SSL(**"imap.gmail.com"**)  
 mail.login(emailid, password)  
 request.session[**'email\_id'**] = emailid  
 request.session[**'password'**] = password  
 sayToUserTTS(**"login successful"**)  
 **return** HttpResponse(json.dumps({**'status'**: 200, **'message'**: **'login successful'**}))  
 **except** Exception **as** e:  
 *# print(e)* sayToUserTTS(**"invalid login credentials or connection error"**)  
 **return** HttpResponse(json.dumps({**'status'**: 401, **'message'**: **'invalid login credentials'**}))  
 **return** render(request, **'login.html'**)  
  
  
*#-------------------------- for sending mail---------------------------------***def** composingEmail(request):  
 sayToUserTTS(**"Composing new Email"**)  
 **return** HttpResponse()  
  
  
**def** receiverEmailId(request):  
 receiver\_emailid = getUserInputSTT(**"say receiver mail id"**)  
 receiver\_emailid = receiver\_emailid.replace(**' '**, **''**).lower().replace(**'dot'**, **'.'**).replace(**'attherate'**, **'@'**)  
 **return** HttpResponse(receiver\_emailid)  
  
  
**def** emailSubject(request):  
 email\_subject = getUserInputSTT(**'say Email subject'**)  
 **return** HttpResponse(email\_subject)  
  
  
**def** textToEmoji(emoji\_text):  
 dataset = pd.read\_csv(**'static/emoji\_dataset.csv'**)  
 data = dataset.loc[dataset[**'cldr\_name'**] == emoji\_text, **'unicode'**].values[0]  
 **return** data.encode(**'utf-8'**).decode(**'unicode\_escape'**).replace(**r'\\U'**, **r'\U'**)  
  
  
**def** emailBody(request):  
 email\_body = **''  
 while True**:  
 type\_body = getUserInputSTT(**'You can type text or emoji in mail body or exit'**)  
 **if 'emoji' in** type\_body:  
 email\_body += **' '**+textToEmoji(getUserInputSTT(**'say Emoji to insert'**))  
 **elif 'text' in** type\_body:  
 email\_body += **' '**+getUserInputSTT(**'say Email message'**)  
 **elif 'exit' in** type\_body:  
 **break  
 return** HttpResponse(email\_body)  
  
**def** sendMail(request):  
 **if** request.method == **'POST'**:  
 email\_id = request.session[**'email\_id'**]  
 password = request.session[**'password'**]  
 receiver\_emailid = request.POST.get(**'receiveremailid'**)  
 email\_subject = request.POST.get(**'emailsubject'**)  
 email\_body = request.POST.get(**'emailbody'**)  
  
 **try**:  
 email\_body\_mime = email.mime.text.MIMEText(email\_body, \_charset=**"UTF-8"**).as\_string()  
 mail = smtplib.SMTP(**'smtp.gmail.com'**, 587) *# host and port area* mail.starttls(context=ssl.create\_default\_context()) *# security connection* mail.login(email\_id, password)  
 mail.sendmail(email\_id, receiver\_emailid, **'Subject: '**+email\_subject+**'\n'**+email\_body\_mime)  
 print(**"Congrates! Your mail has send. "**)  
 sayToUserTTS(**'your mail is sent'**)  
 **return** HttpResponse(json.dumps({**'status'**: 200, **'message'**: **'mail sent'**}))  
 **except**:  
 sayToUserTTS(**'oops unable to send mail something went wrong'**)  
 **return** HttpResponse(json.dumps({**'status'**: 400, **'message'**: **'something went wrong'**}))  
 **return** render(request, **'sendmail.html'**)  
  
  
*#-------------------------- for reading mails---------------------------------***def** readMails(request):  
 **try**:  
 context\_dict = {}  
 email\_id = request.session[**'email\_id'**]  
 password = request.session[**'password'**]  
  
 mail = imaplib.IMAP4\_SSL(**"imap.gmail.com"**)  
 mail.login(email\_id, password)  
 read\_mails\_flag = getUserInputSTT(**'read all or unseen mails from inbox or read sent mails'**)  
 start\_reading = 0  
 **if 'all' in** read\_mails\_flag **or 'unseen' in** read\_mails\_flag:  
 read\_mails\_flag = **'ALL' if 'all' in** read\_mails\_flag.lower() **else 'UNSEEN'** mail.select(**"INBOX"**)  
 status, messages = mail.search(**None**, read\_mails\_flag)  
  
 **if** read\_mails\_flag == **'ALL'**:  
 messages=[messages[0].split()[-1]]  
 **if** read\_mails\_flag == **'UNSEEN'**:  
 temp=messages[0].split()  
 start\_reading = int(temp[0])-1  
 messages[0] = temp[-1]  
 **elif 'sent' in** read\_mails\_flag:  
 status, messages = mail.select(**'"[Gmail]/Sent Mail"'**)  
  
 messages\_count = int(messages[0])  
 **for** i **in** range(messages\_count, start\_reading, -1):  
 typ, data = mail.fetch(str(i), **'(RFC822)'**)  
 **for** response\_part **in** data:  
 **if** isinstance(response\_part, tuple):  
 msg = email.message\_from\_string(response\_part[1].decode(**"utf-8"**))  
 email\_subject = str(msg[**'subject'**])  
 email\_from = str(msg[**'from'**])  
 body = **''  
 if** msg.is\_multipart():  
 **for** part **in** msg.walk():  
 **if** part.is\_multipart():  
 **for** subpart **in** part.get\_payload():  
 **if** subpart.is\_multipart():  
 **for** subsubpart **in** subpart.get\_payload():  
 body = body + str(subsubpart.get\_payload(decode=**True**)) + **'\n'  
 else**:  
 body = body + str(subpart.get\_payload(decode=**True**)) + **'\n'  
 else**:  
 body = body + str(part.get\_payload(decode=**True**)) + **'\n'  
 else**:  
 body = body + str(msg.get\_payload(decode=**True**)) + **'\n'** body = bytes(body, **'utf-8'**).decode(**'unicode-escape'**)  
  
 **import** html2text  
 h = html2text.HTML2Text()  
 h.ignore\_links = **True** email\_body = h.handle(body)  
 **import** re  
 email\_body = re.sub(**r'(https|http)?:\/\/(\w|\.|\/|\?|\=|\&|\%)\*\b'**, **''**,  
 email\_body.split(**' b'**, 1)[0], flags=re.MULTILINE)  
 email\_body = email\_body[2: len(email\_body) - 1:]  
 context\_dict[i] = {**'from'**: email\_from, **'subject'**: email\_subject, **'body'**: email\_body}  
 **except** Exception **as** e:  
 sayToUserTTS(**'oops unable to read mails'**)  
  
 readMailsContext(context\_dict)  
 request.session[**'processnext'**] = threading.active\_count()  
 **return** render(request,**'readmails.html'**, {**'data'**: context\_dict})  
  
**def** processNext(request):  
 **while True**:  
 **if** threading.active\_count() < request.session[**'processnext'**]:**break  
 else**:time.sleep(1)  
 **return** HttpResponse(json.dumps({**'status'**:200,**'processnext'**:**'completed'**}))  
  
**def** logout(request):  
 **del** request.session[**'email\_id'**]  
 **del** request.session[**'password'**]  
 **del** request.session[**'processnext'**]  
 **return** redirect(**'login'**)  
  
**def** aboutUs(request):  
 **return** render(request,**'aboutus.html'**)

**SCREEN SHOTS:**

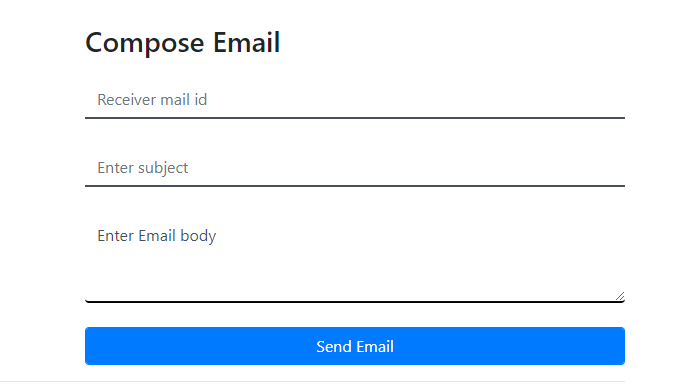
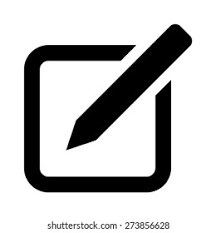
**Home page:**



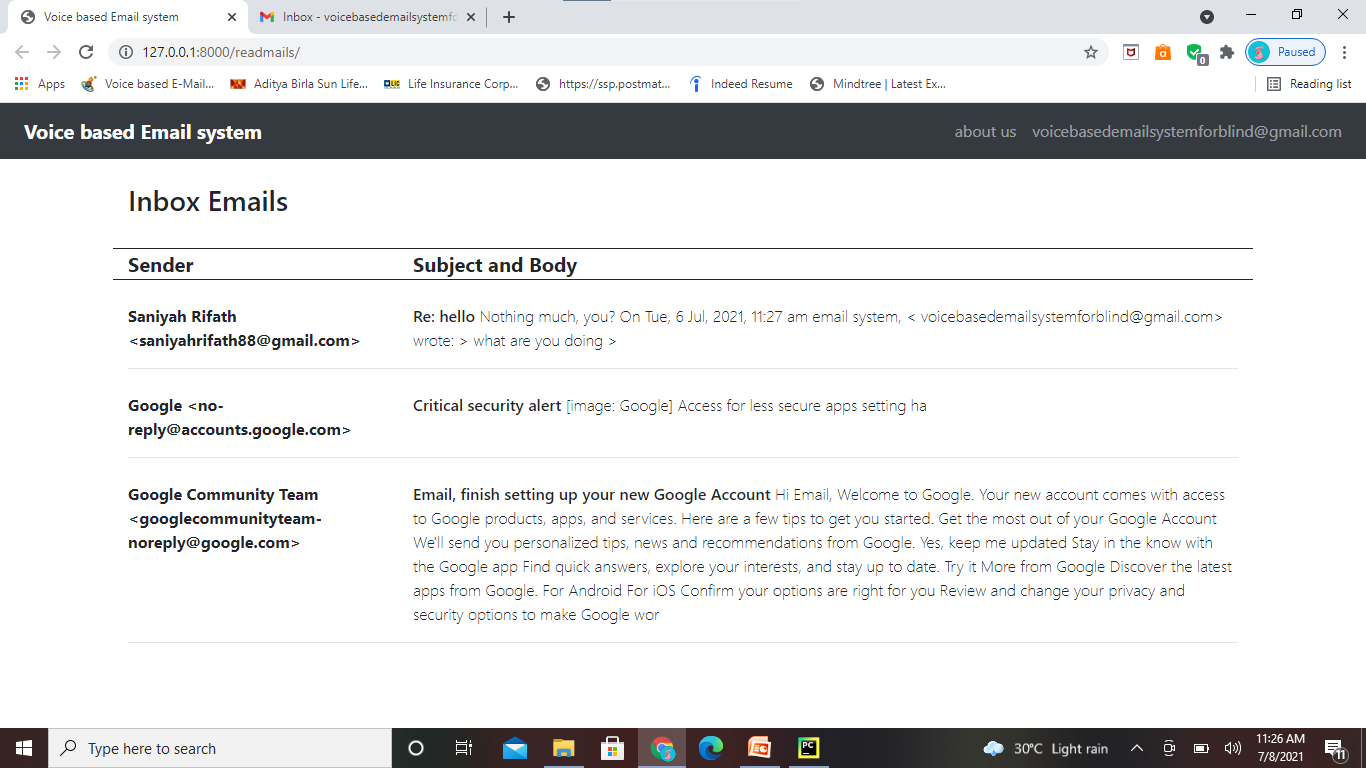
**Login Page:**



**Compose Mail Page:**



**Inbox Page:**



1. **CONCLUSION**

This e-mail system can be used by any user of any age group with ease of access. It has feature of speech to text as well as text to speech with speech reader which makes designed system to be handled by visually impaired person as well as blind person.

This system will help in overcoming some drawbacks that were earlier faced by the blind people in accessing emails. We have eliminated the concept of using keyboard shortcuts along with screen readers which will help reducing the cognitive load of remembering keyboard shortcuts.

Also any non-sophisticated user who does not know the position of keys on the keyboard need not bother as keyboard usage is eliminated. Instructions given by the IVR accordingly to get the respective services offered.

Other than this the user might need to feed in information through voice inputs when specified.

It is a observation that about 70% of total blind population across the world is present in INDIA. This project, describe the voice mail architecture used by blind people to access Email 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. It also helps handicapped and illiterate people.

1. **FUTURE SCOPE**

There is wide future scope of this system many enhancements can be done in the system such as including different languages, including functionality of accessing the deleted mails and spam mails. This system can be made available to all regional people who are not educated enough and inclusion of different languages will make this system easily accessible. Furthermore sign language system can also be integrated with the system to make the system more scalable and robust.

1. **REFERENCES**

[1] T.shabana, a.anam, a.rafiya3, k.aisha, “voice based email system for blinds” <http://www.ijarcce.com/upload/2015/january/ijarcce5c.pdf>

[2]Codeproject,“speechrecognition” <http://www.codeproject.com/articles/5820/speechrecognition>

[3] Ummuhanysifa u., nizarbanu p k, “voice based search engine and web page reader”. In International journal of computational engineering research (ijcer).

[4] <Http://www.ijceronline.com/papers/special%20issue/a0105.pdf>

[5] Arjun aj, “voice based email for blinds”, slide share

http://www.slideshare.net/123arjun1 /voicebased-email-for-blinds