Minor Project Lab

Project Report

ON

Voice Based E-mail For Blind People

BY:

ADITYA PAL (E20166135500004)

AKASH SINGH (E20166135500009)

PRINCE SHARMA (E2016613550045)

Department of Computer Science & Engineering,

**Government Polytechnic, Mawana Khurd,
Meerut**

Session:- 2022-23

CANDIDATE’S DECLARATION

I/we hereby declare that the project entitled **“Voice based email for blind people”** submitted in partial fulfillment for requirements of the Minor Project Lab completed under the guidance of **Mr. Pankaj Sharma, Lecturer,** Government Polytechnic, Mawana Khurd, Meerut is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award/ requirements of any other degree/course elsewhere.

Signature & name of the student(s) with date.

Certificate By the Project Guide

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of the project guide with dates and their designation.

Abstract

Internet is one of the basic need for daily living. Every person is using the facts and information on internet. On the other hand, blind people face problem in accessing the text resources. The advancement in computer based accessible systems has opened up many ways 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 the visually challenged people find it very difficult to utilize this technology because of the fact that using them requires them to see. In this project, the voicemail system architecture that can be used by a blind person to access e-mails easily and efficiently. The contribution made by project has helped the blind people to send and receive voice- based e-mail message. Also this system can be used by any normal person also for example the one who is not able to read. The system is completely based on interactive voice response which will make it efficient.

Table of Contents

1. Introduction

E-mails are the most common way of communication over Internet, for sending and receiving some important information. But there is a certain way for humans to access the Internet and the criteria 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. The only way by which a visually challenged person can send an E-mail is, they have to speak out loud the entire content of the mail to another person (not visually challenged) and then that third person will compose the mail and sendon the behalf of the visually challenged person. But this is not a correctway to deal with the scenario. It is very unlikely that every time a blindperson can find someone for help. So, for the betterment of society and giving an equal status to such specially able people we have comeup with this project idea which provides the user with ability to send mails using voice commands without the need of keyboard or any other thing just their voice.

There are various similar and advanced project on this topic made by different developers which in some manner their are almost similar to our project as the root cause is to make the blind people send their emails ,check their inbox etc. and various functions. The project which we are making is a very basic version of the advanced projects which are available in the internet . In our project , system will ask the userfor its login credentials . Then it will give user some options to perform ,user have to select the option and give the required details/informations .user will be informed in the end as the operation is done . All this process will be going on through voice. The system will guide the user through voice throughout the program and the user will also have to respond to the commands in voice only .

2. Problem Formulation

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 .

3. Related Work

3.1 “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. this paper describes, 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 the mental load taken by blind to remember and type characters using keyboard. This includes development of text to Braille systems, screen magnifiers and screen readers.

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

This paper describes 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.

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

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. In this system mainly three types of technologies are used namely: STT (Speech-to-text), TTS (text-to-speech) ,IVR (Interactive voice response).

3.4 “T. Shabana, A. Anam, A.Rafiya, K. Aisha, "Voice based Email System for Blinds", "IJARCCE", Jan i2015” The authors propose theTetra-Entry, a blind friendly email client, to overcome the favourability and the convenience of email activities.

3.5 Rijwan Khan, Pawan Kumar Sharma, Sumit Raj, Sushil Kr. Verma, Sparsh Katiyar. "Voice Based EMail System using Artificial Intelligence". International Journal of Engineering and Advanced Technology (IJEAT) ISSN: i2249– i8958, Volume-9 Issue-3, February, i2020.A system for the blind and the illiterate is proposed to improve their interaction with the email system. This system eliminates the use of IVR technology that are using screen readers, braille keyboards. There, we used the speech-to-text and text-to- speech conversion. Voice commands are also used for other activities as well

4. Research Methodology

4.1 Overview

This Project will do the following things:

- Ask the user for their login credentials
- Provide user two options – 1. compose 2. Check inbox
- Compose will send the email.
- Check inbox will show the recent emails received.
- All interactions will be through voice only. No use of keyboard andmouse is required.

4.2 Requirements

Basic requirements to make this projects are:

- Any python interpreter like vs code, spider , anaconda etc.
- Due to more security purpose, make third party app password forsmtplib and imaplib and then use it as the email login password .
- Install the required modules :
  1. Speechrecognition
  2. Smtplib
  3. Imaplib
  4. gtts

  5. Beautifulsoup
  6.Pyglet
  7. Email
  8.Time

4.3 Basic information of modules

1. SpeechRecognition

Speech Recognition module basically converts the spoken word to text. Python supports many speech recognition engines and APIs ,including Google Speech Engine ,Google Cloud Speech API etc.

2. gtts

This is a module which is developed by the Google and contributed in the major part in the project that is converting text to speech.

3.pyglet

It’s a cross platform windowing and multimedia library. This library was used to play converted text files. All the voice that we listen in this project was made possible by this library.

4.smtplib

The smtp module defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP or ESMTP listener daemon. This module is heart of the project. Using this we were able to send mail. .

5. Beautifulsoup4

Beautiful Soup is a library that makes it easy to scrape informationfrom web pages.

6. imaplib

IMAP is an email retrieval protocol which does not download the emails. It just reads them and displays them. This is very useful in low bandwidth condition. Python’s client side library called **imaplib** is used for accessing emails over imap protocol.

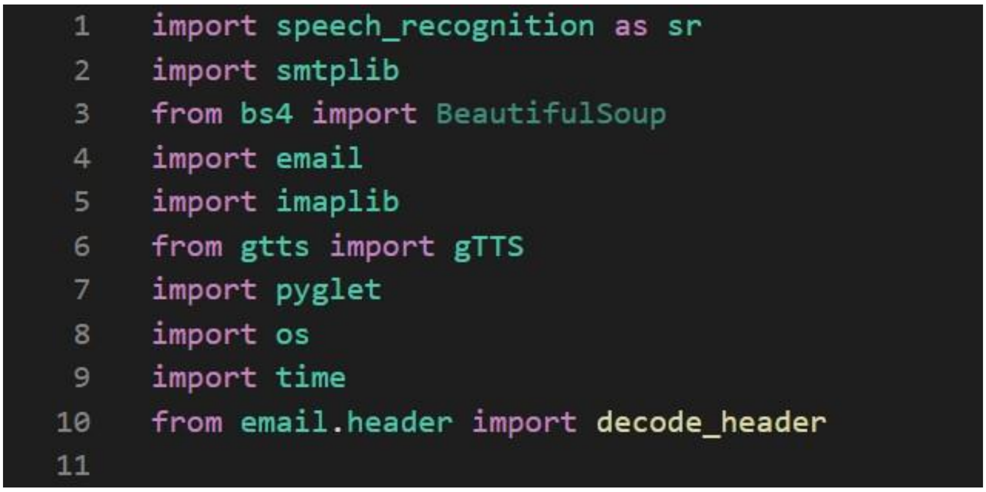
7. OS module

The os module in python provides functions for interacting with the operating system. OS comes under python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality. We’ve implemented this module to delete the files that are no longer in use.

8. Time module

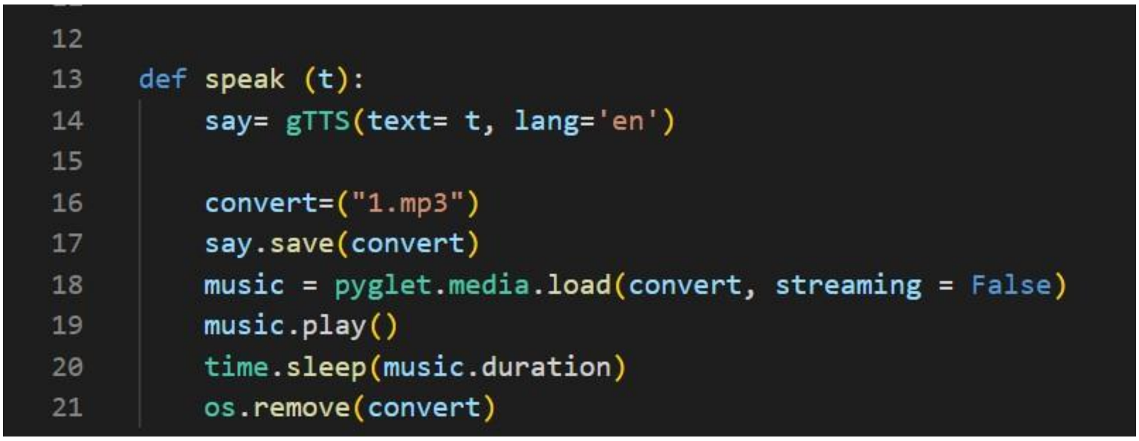
We’ve used this module to stop the execution of the code until the user input it’s voice.

Imported Modules



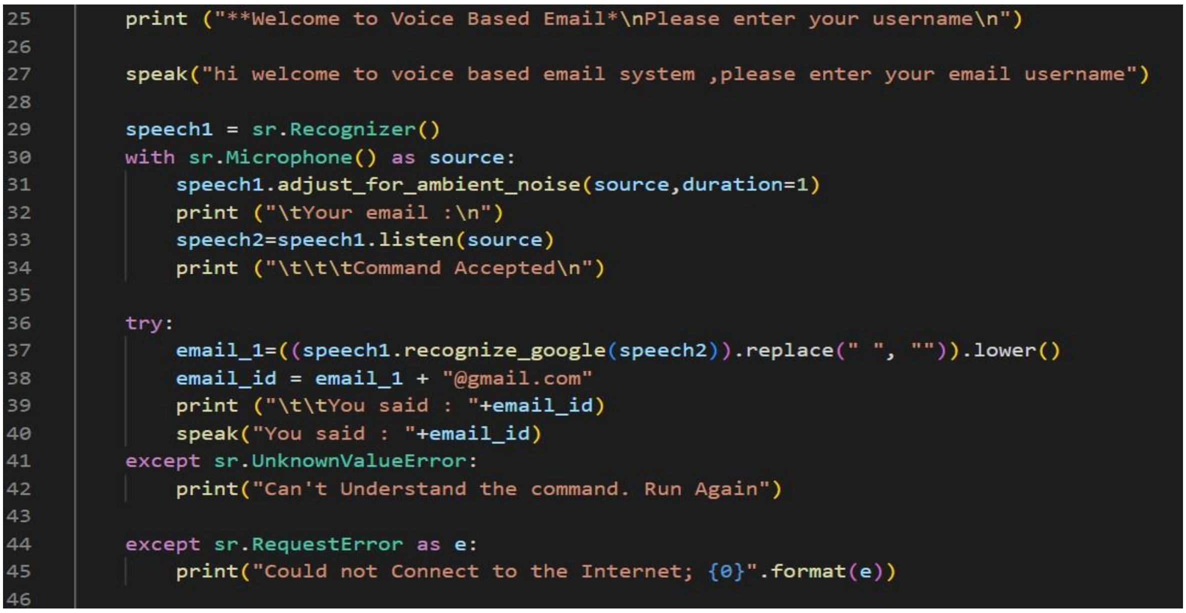
```
import speech_recognition as sr
import smtplib
from bs4 import BeautifulSoup
import email
import imaplib
from gtts import gTTS
import pyglet
import os
import time
from email.header import decode_header

```

Code for text to speech

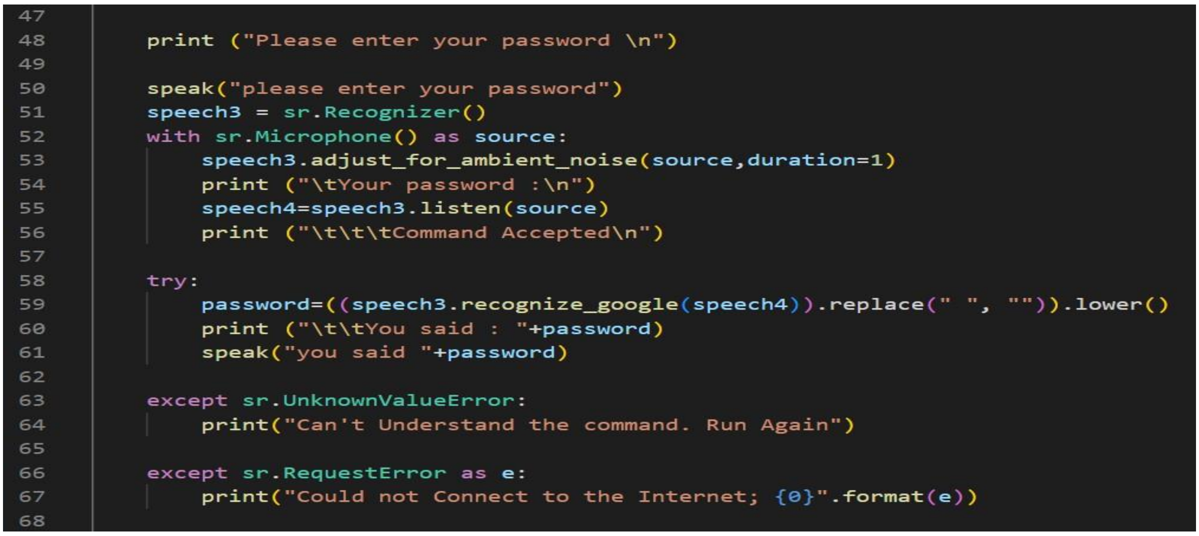


```python
def speak (t):
    say= gTTS(text= t, lang='en')

    convert=("1.mp3")
    say.save(convert)
    music = pyglet.media.load(convert, streaming = False)
    music.play()
    time.sleep(music.duration)
    os.remove(convert)
```

Code for login create details

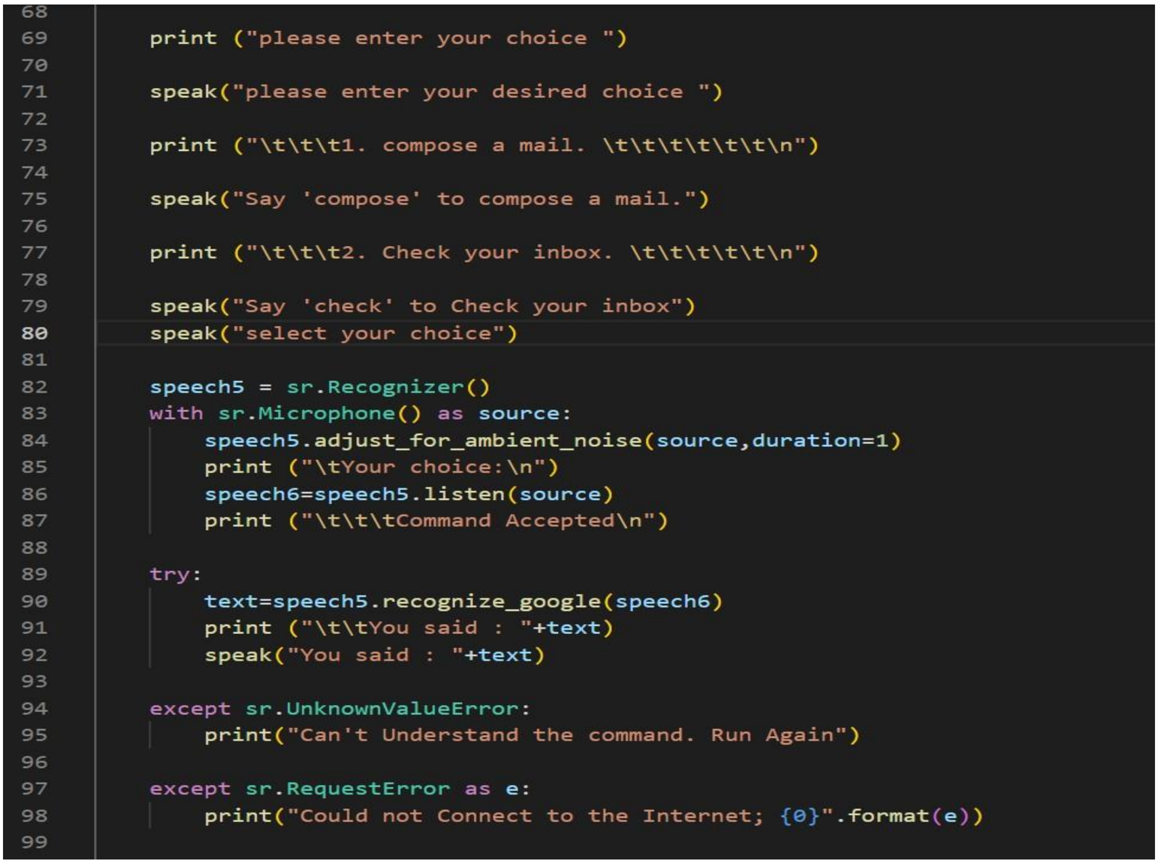


```python
    print ("**Welcome to Voice Based Email*\nPlease enter your username\n")

    speak("hi welcome to voice based email system ,please enter your email username")

    speech1 = sr.Recognizer()
    with sr.Microphone() as source:
        speech1.adjust_for_ambient_noise(source,duration=1)
        print ("\tYour email :\n")
        speech2=speech1.listen(source)
        print ("\t\t\tCommand Accepted\n")

    try:
        email_1=((speech1.recognize_google(speech2)).replace(" ", "")).lower()
        email_id = email_1 + "@gmail.com"
        print ("\t\tYou said : "+email_id)
        speak("You said : "+email_id)
    except sr.UnknownValueError:
        print("Can't Understand the command. Run Again")

    except sr.RequestError as e:
        print("Could not Connect to the Internet; {0}".format(e))

```



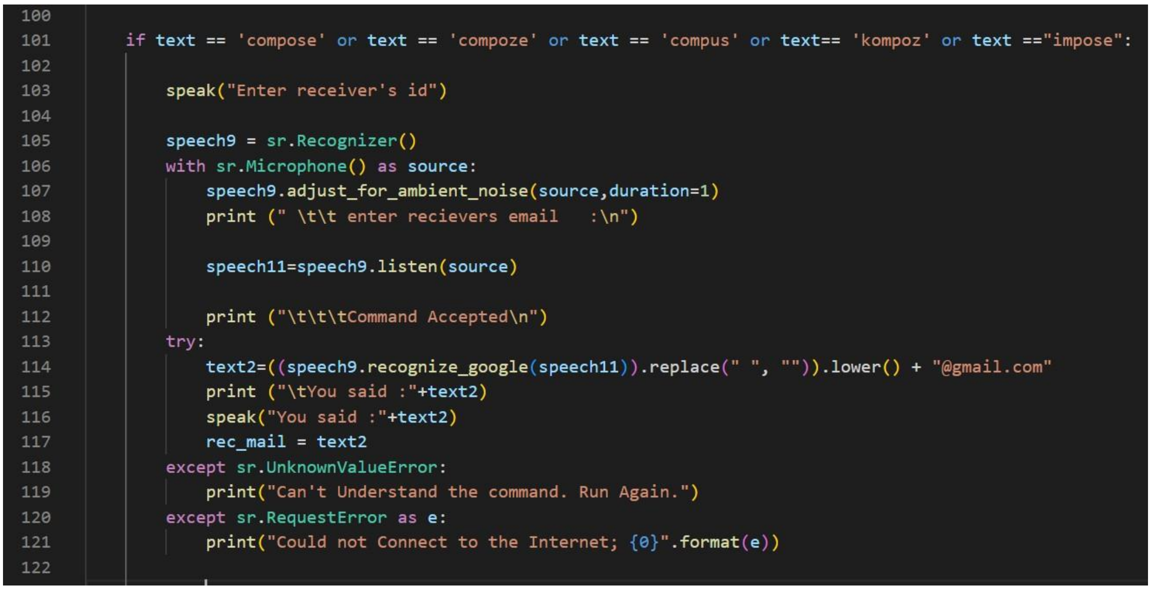
```python

    print ("Please enter your password \n")

    speak("please enter your password")
    speech3 = sr.Recognizer()
    with sr.Microphone() as source:
        speech3.adjust_for_ambient_noise(source,duration=1)
        print ("\tYour password :\n")
        speech4=speech3.listen(source)
        print ("\t\t\tCommand Accepted\n")

    try:
        password=((speech3.recognize_google(speech4)).replace(" ", "")).lower()
        print ("\t\tYou said : "+password)
        speak("you said "+password)

    except sr.UnknownValueError:
        print("Can't Understand the command. Run Again")

    except sr.RequestError as e:
        print("Could not Connect to the Internet; {0}".format(e))

```

Code for choices

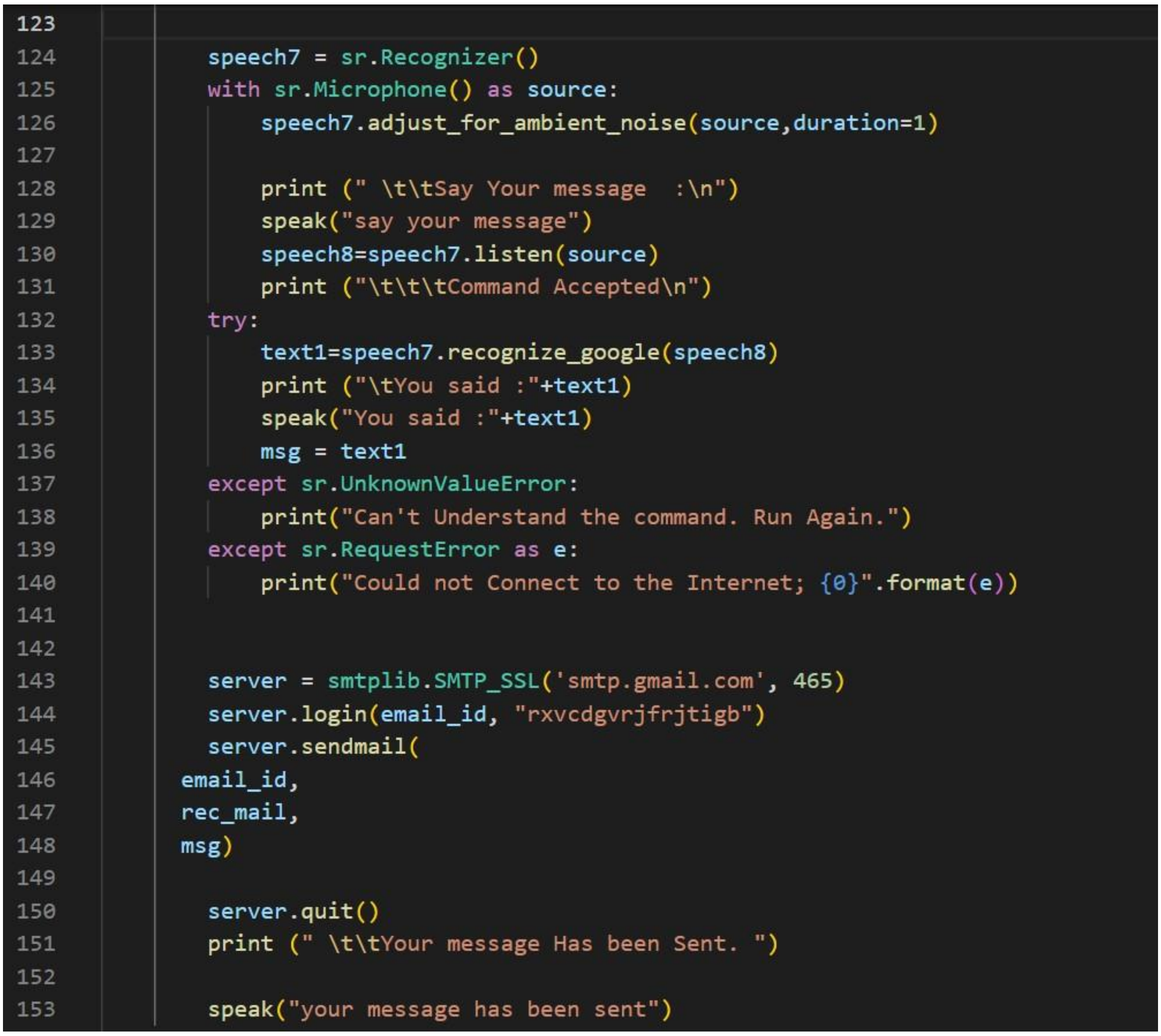


```python
    print ("please enter your choice ")

    speak("please enter your desired choice ")

    print ("\t\t\t1. compose a mail. \t\t\t\t\t\t\n")

    speak("Say 'compose' to compose a mail.")

    print ("\t\t\t2. Check your inbox. \t\t\t\t\t\n")

    speak("Say 'check' to Check your inbox")
    speak("select your choice")

    speech5 = sr.Recognizer()
    with sr.Microphone() as source:
        speech5.adjust_for_ambient_noise(source,duration=1)
        print ("\tYour choice:\n")
        speech6=speech5.listen(source)
        print ("\t\t\tCommand Accepted\n")

    try:
        text=speech5.recognize_google(speech6)
        print ("\t\tYou said : "+text)
        speak("You said : "+text)

    except sr.UnknownValueError:
        print("Can't Understand the command. Run Again")

    except sr.RequestError as e:
        print("Could not Connect to the Internet; {0}".format(e))

```

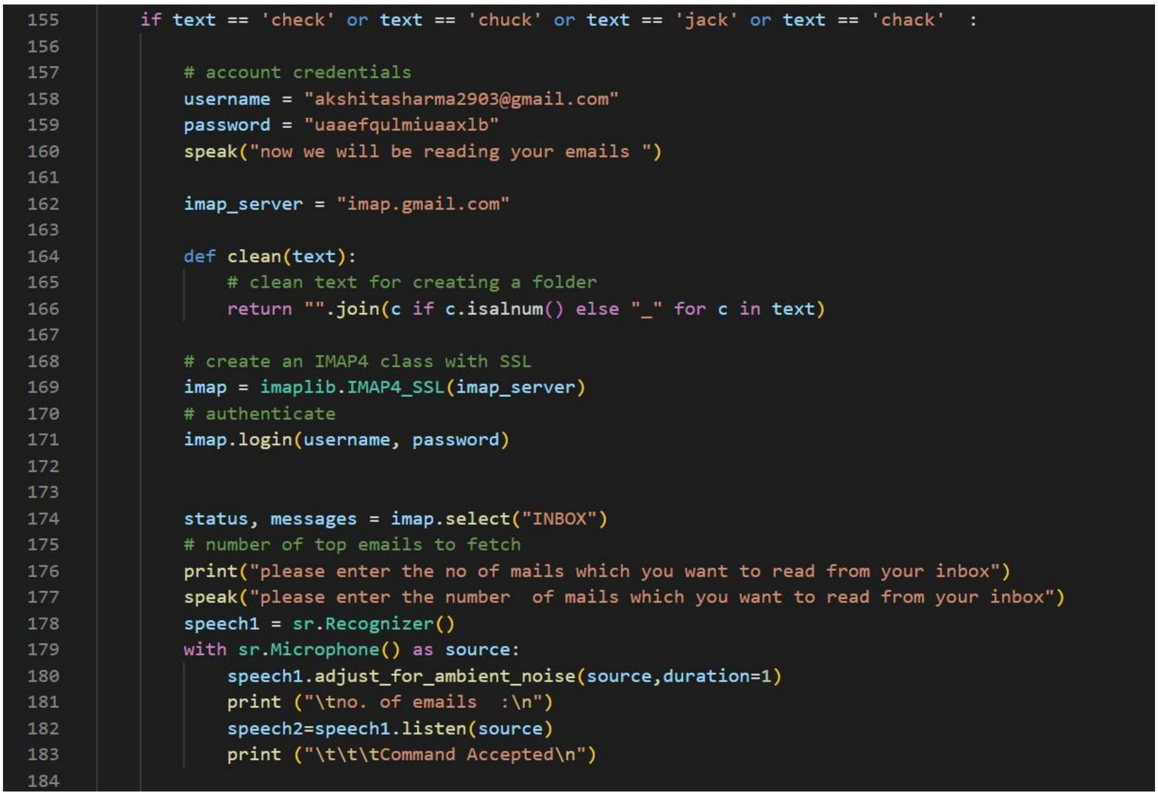
Code for compose and checking inbox



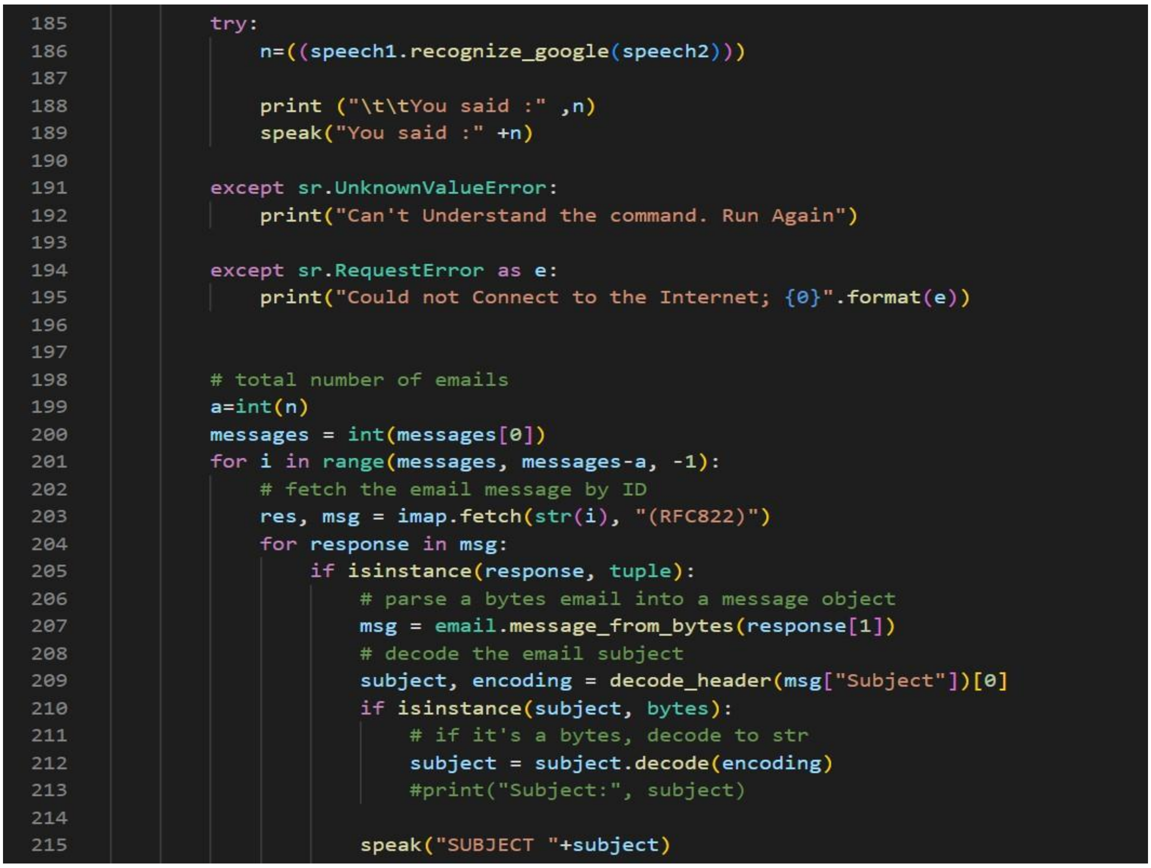
```python

    if text == 'compose' or text == 'compoze' or text == 'compus' or text== 'kompoz' or text =="impose":

        speak("Enter receiver's id")

        speech9 = sr.Recognizer()
        with sr.Microphone() as source:
            speech9.adjust_for_ambient_noise(source,duration=1)
            print (" \t\t enter recievers email   :\n")

            speech11=speech9.listen(source)

            print ("\t\t\tCommand Accepted\n")
        try:
            text2=((speech9.recognize_google(speech11)).replace(" ", "")).lower() + "@gmail.com"
            print ("\tYou said :"+text2)
            speak("You said :"+text2)
            rec_mail = text2
        except sr.UnknownValueError:
            print("Can't Understand the command. Run Again.")
        except sr.RequestError as e:
            print("Could not Connect to the Internet; {0}".format(e))

```



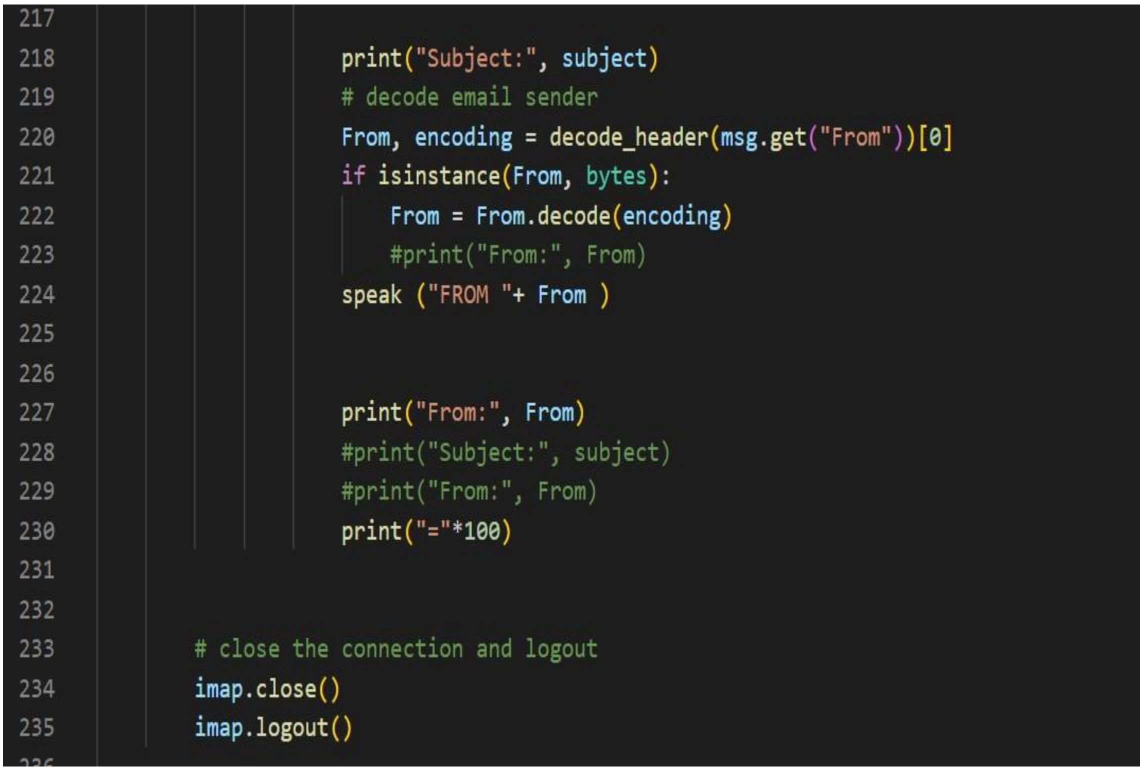
```python

    speech7 = sr.Recognizer()
    with sr.Microphone() as source:
        speech7.adjust_for_ambient_noise(source,duration=1)

        print (" \t\tSay Your message  :\n")
        speak("say your message")
        speech8=speech7.listen(source)
        print ("\t\t\tCommand Accepted\n")
    try:
        text1=speech7.recognize_google(speech8)
        print ("\tYou said :"+text1)
        speak("You said :"+text1)
        msg = text1
    except sr.UnknownValueError:
        print("Can't Understand the command. Run Again.")
    except sr.RequestError as e:
        print("Could not Connect to the Internet; {0}".format(e))


    server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    server.login(email_id, "rxvcdgvrjfrjtigb")
    server.sendmail(
  email_id,
  rec_mail,
  msg)

    server.quit()
    print (" \t\tYour message Has been Sent. ")

    speak("your message has been sent")
```



```python
    if text == 'check' or text == 'chuck' or text == 'jack' or text == 'chack'  :

        # account credentials
        username = "akshitasharma2903@gmail.com"
        password = "uaaefqulmiuaaxlb"
        speak("now we will be reading your emails ")

        imap_server = "imap.gmail.com"

        def clean(text):
            # clean text for creating a folder
            return "".join(c if c.isalnum() else "_" for c in text)

        # create an IMAP4 class with SSL
        imap = imaplib.IMAP4_SSL(imap_server)
        # authenticate
        imap.login(username, password)


        status, messages = imap.select("INBOX")
        # number of top emails to fetch
        print("please enter the no of mails which you want to read from your inbox")
        speak("please enter the number  of mails which you want to read from your inbox")
        speech1 = sr.Recognizer()
        with sr.Microphone() as source:
            speech1.adjust_for_ambient_noise(source,duration=1)
            print ("\tno. of emails  :\n")
            speech2=speech1.listen(source)
            print ("\t\t\tCommand Accepted\n")

```

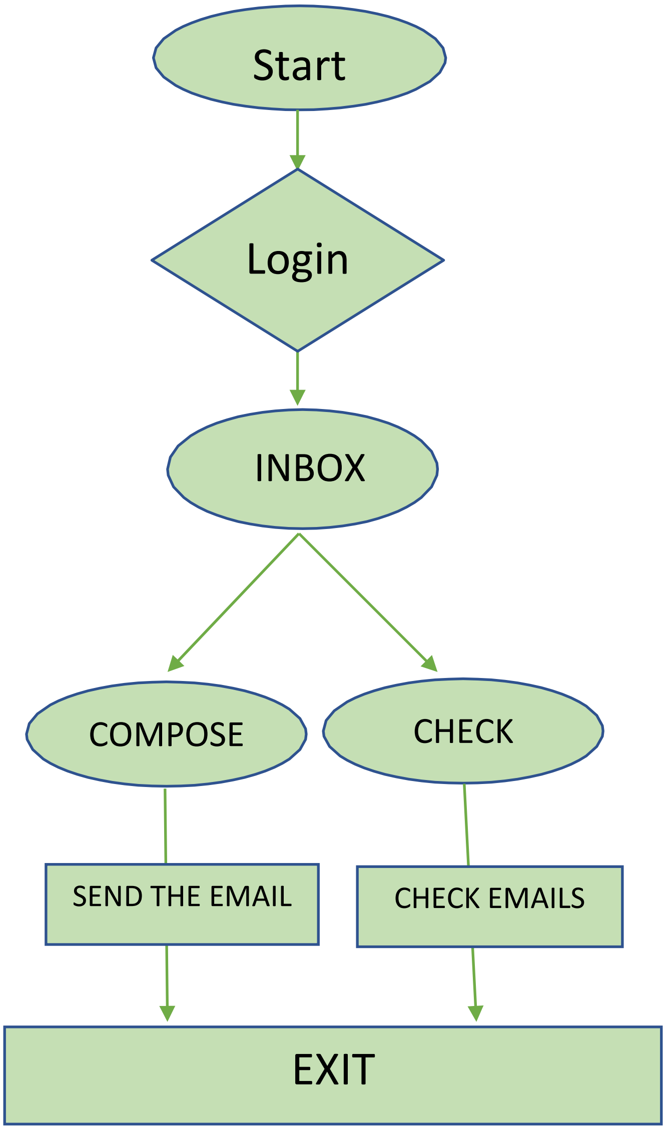


```
            try:
                n=((speech1.recognize_google(speech2)))

                print ("\t\tYou said :" ,n)
                speak("You said :" +n)

            except sr.UnknownValueError:
                print("Can't Understand the command. Run Again")

            except sr.RequestError as e:
                print("Could not Connect to the Internet; {0}".format(e))


            # total number of emails
            a=int(n)
            messages = int(messages[0])
            for i in range(messages, messages-a, -1):
                # fetch the email message by ID
                res, msg = imap.fetch(str(i), "(RFC822)")
                for response in msg:
                    if isinstance(response, tuple):
                        # parse a bytes email into a message object
                        msg = email.message_from_bytes(response[1])
                        # decode the email subject
                        subject, encoding = decode_header(msg["Subject"])[0]
                        if isinstance(subject, bytes):
                            # if it's a bytes, decode to str
                            subject = subject.decode(encoding)
                            #print("Subject:", subject)

                        speak("SUBJECT "+subject)
```



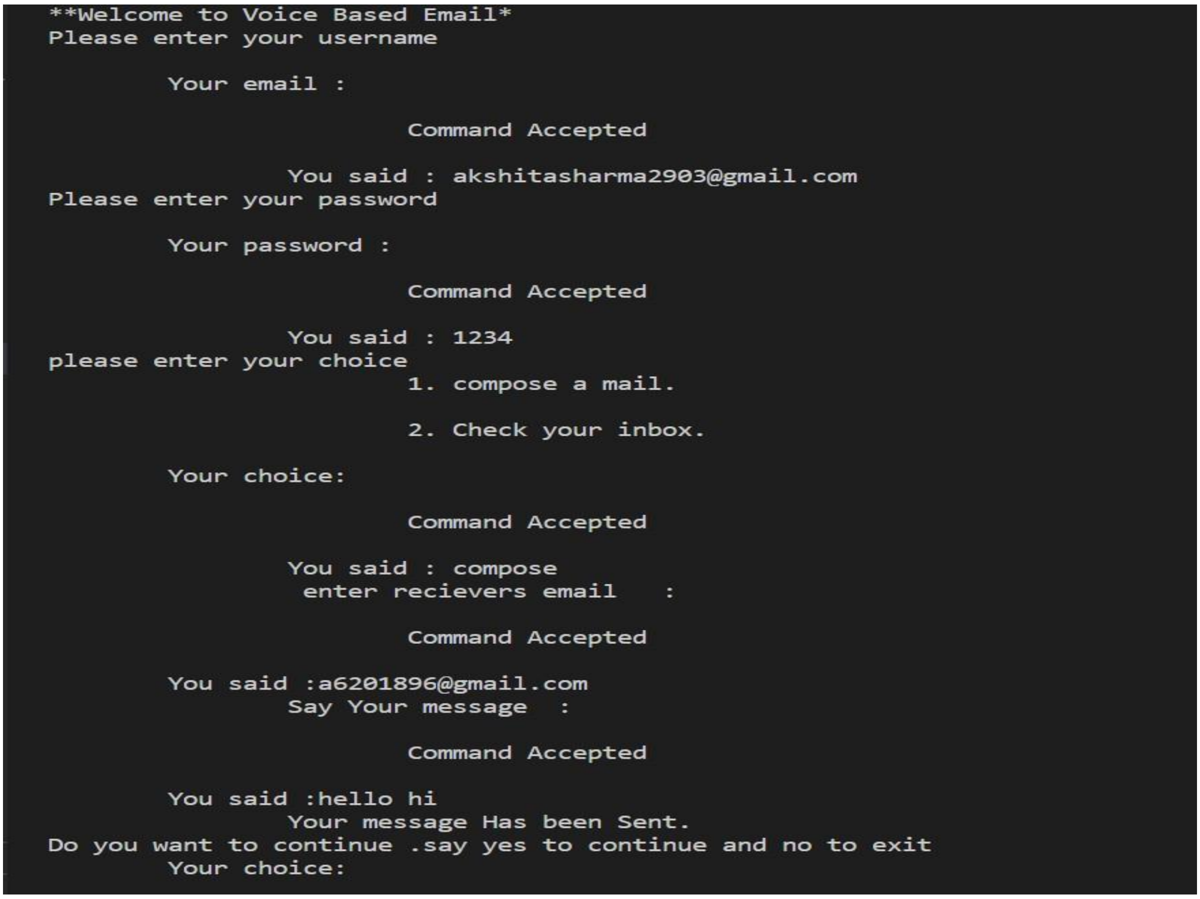
```

                        print("Subject:", subject)
                        # decode email sender
                        From, encoding = decode_header(msg.get("From"))[0]
                        if isinstance(From, bytes):
                            From = From.decode(encoding)
                            #print("From:", From)
                        speak ("FROM "+ From )


                        print("From:", From)
                        #print("Subject:", subject)
                        #print("From:", From)
                        print("="*100)


            # close the connection and logout
            imap.close()
            imap.logout()
```

Flowchart

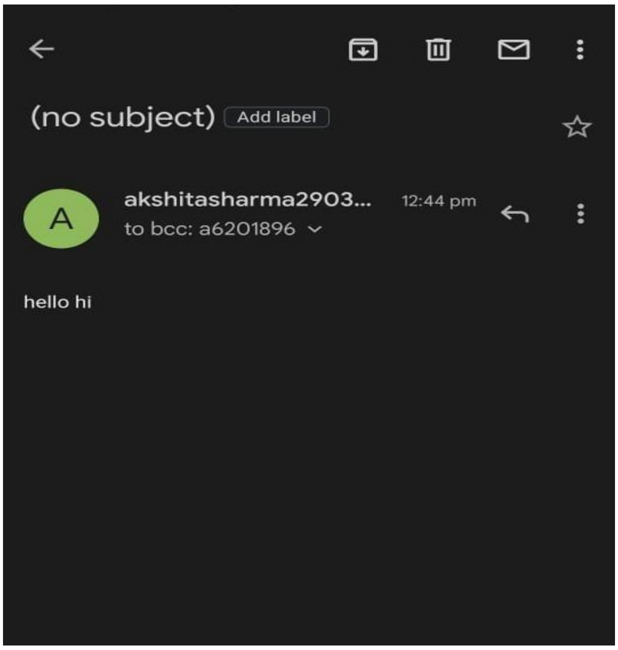

Start
Login
INBOX
COMPOSE
CHECK
SEND THE EMAIL
CHECK EMAILS
EXIT

5. Experimental Results

Output for compose mail



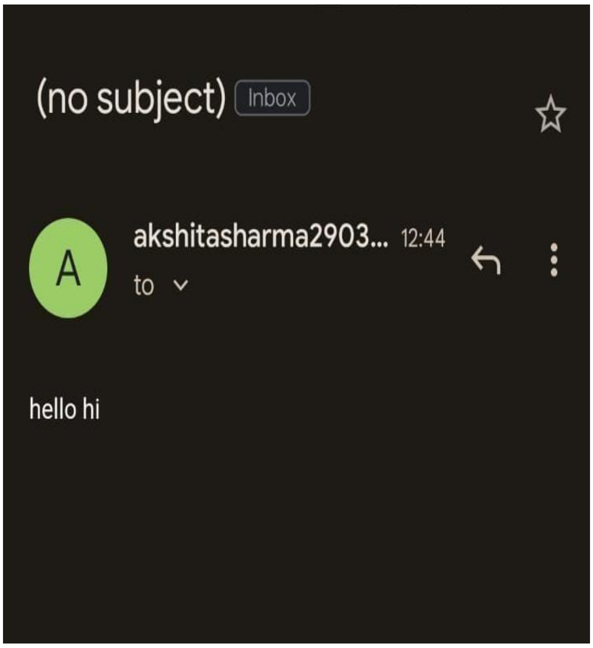
```
**Welcome to Voice Based Email*
Please enter your username

        Your email :

                        Command Accepted

                You said : akshitasharma2903@gmail.com
Please enter your password

        Your password :

                        Command Accepted

                You said : 1234
please enter your choice
                        1. compose a mail.

                        2. Check your inbox.

        Your choice:

                        Command Accepted

                You said : compose
                 enter recievers email    :

                        Command Accepted

        You said :a6201896@gmail.com
                Say Your message  :

                        Command Accepted

        You said :hello hi
                Your message Has been Sent.
Do you want to continue .say yes to continue and no to exit
        Your choice:
```



akshitasharma2903... 12:44 pm

to bcc: a6201896

hello hi

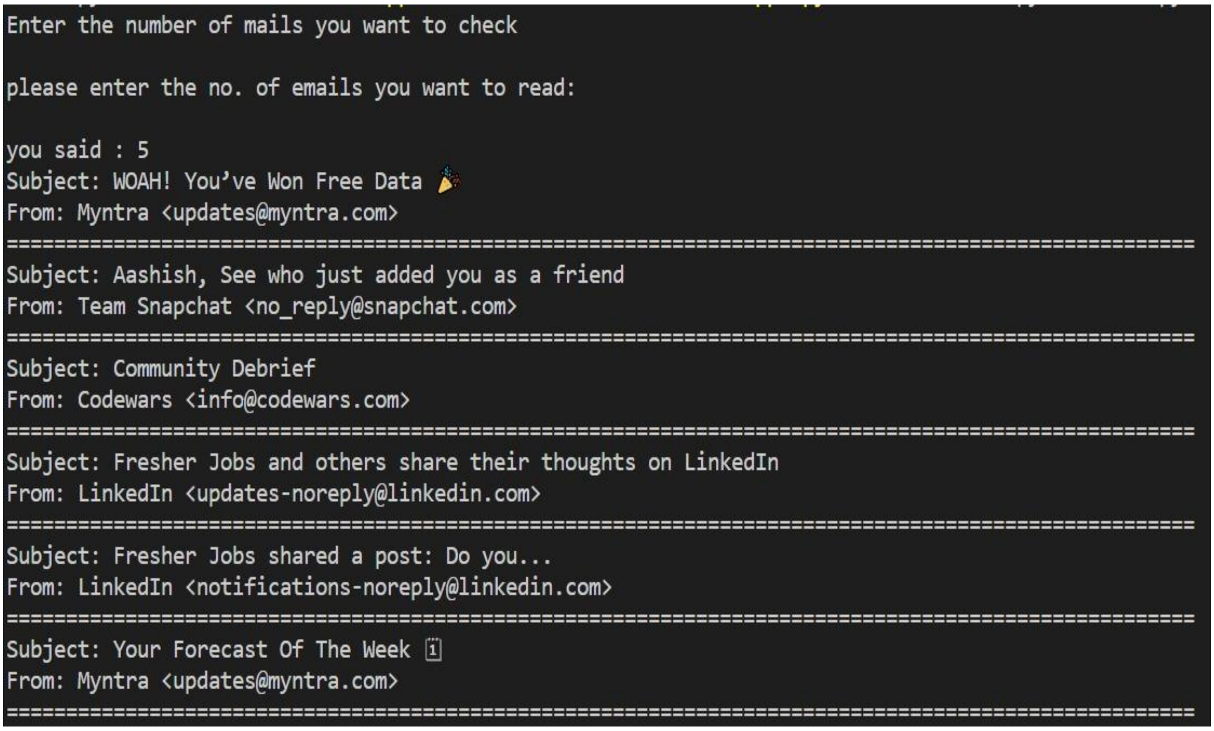


akshitasharma2903... 12:44

to

hello hi

Output to check email



```
Enter the number of mails you want to check

please enter the no. of emails you want to read:

you said : 5
Subject: WOAH! You’ve Won Free Data 🎉
From: Myntra <updates@myntra.com>
====================================================================================================
Subject: Aashish, See who just added you as a friend
From: Team Snapchat <no_reply@snapchat.com>
====================================================================================================
Subject: Community Debrief
From: Codewars <info@codewars.com>
====================================================================================================
Subject: Fresher Jobs and others share their thoughts on LinkedIn
From: LinkedIn <updates-noreply@linkedin.com>
====================================================================================================
Subject: Fresher Jobs shared a post: Do you...
From: LinkedIn <notifications-noreply@linkedin.com>
====================================================================================================
Subject: Your Forecast Of The Week 1️⃣
From: Myntra <updates@myntra.com>
====================================================================================================
```

6. Conclusion and Future Work

This project suggests that it will be useful for the community so thatpeople with disabilities to develop on the side of the village. This project allows people with visual impairments to be quite capable ofbeing a part of the growth of digital India and on their ability to communicate over the Internet and in people's lives easier. This system overcomes many of the disadvantages of the people down your face when you see how to send and receive email. The successof this project can affect the developers, encouraging them to createuseful products that can help people with low vision or who are blind. In the future, this system has a lot of potential, with a lot of enhancements; perhaps it was a system that, simply send email andcheck the inbox. In addition, this system can be upgraded to send an attachment, more beautification can be done for this project like making an app or web application with the same concept. More features can be added to this like changing password, changing username, searching for a mail, checking spam folder etc. just like the actual email do. This project could be a gem for the society.

Reference

[1]. Guillermo Arturo Hernández Tapia, Ana Lilia Reyes Herrera. "E- mail management system for blind people in Spanish language". In Interaccióni'17: XVIII International Conference on Human Computer Interaction Cancun Mexico September, i2017.

[2]. Jagtap Nilesh, Pawan Alai, Chavhan Swapnil, and Bendre M.R "Voice Based System in Desktop and Mobile Devices for Blind People". In International Journal of Emerging Technology and Advanced Engineering (IJETAE), i2014 ion Pages i404-407.

[3]. Payal Dudhbale J. S.Wankhade, P. S. Narawade ."Voice-Based System in Desktop and Mobile Devices for Blind People ". In International Journal of Scientific Research in Science and Technology,i2018.

[4]. Ruchi Khedekar, Sonu Gupta, i2019, Voice based email System for Blinds, INTERNATIONAL JOURNALiOFiENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume i08, Issue i10 (October i2019).

[5]. G. Shoba, G. Anusha, V.Jeevitha, R.Shanmathi."AN Interactive Email for Visually Impaired". In International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), i2014 ion Pages i5089-5092 (Volume i3, Issue i1).

[6]. Rijwan Khan, Pawan Kumar Sharma, Sumit Raj, Sushil Kr. Verma, Sparsh Katiyar. "Voice Based EMail System using Artificial Intelligence". International Journal of Engineering and Advanced Technology (IJEAT) ISSN: i2249– i8958, Volume-9 Issue-3, February, i2020.

[7]. Runze Chen, Zhanhong Tian, Hailun Liu, Fang Zhao, Shuai Zhang, HaoboiLiu "Construction of a Voice Driven Life Assistant System for Visually Impaired People "International Conference on Artificial Intelligence and Big Data-" IEEE, i2018, PP i87-92, ISSN i5386-6987.

[8]. T. Shabana, A. Anam, A.Rafiya, K. Aisha, "Voice based Email System for Blinds", "IJARCCE", Jan i2015.

[9]. Jayachandran, K., & Anbumani, P. (2017). Voice based email for blind people. Int. J. Adv. Res. Ideas Innov. Technol., i3(3), i1065-1071.

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