***Project* *Report***

***On***

**VOICE BASED EMAIL SERVICE**

*(CSE IV Semester Mini project)*

*2020-2021*



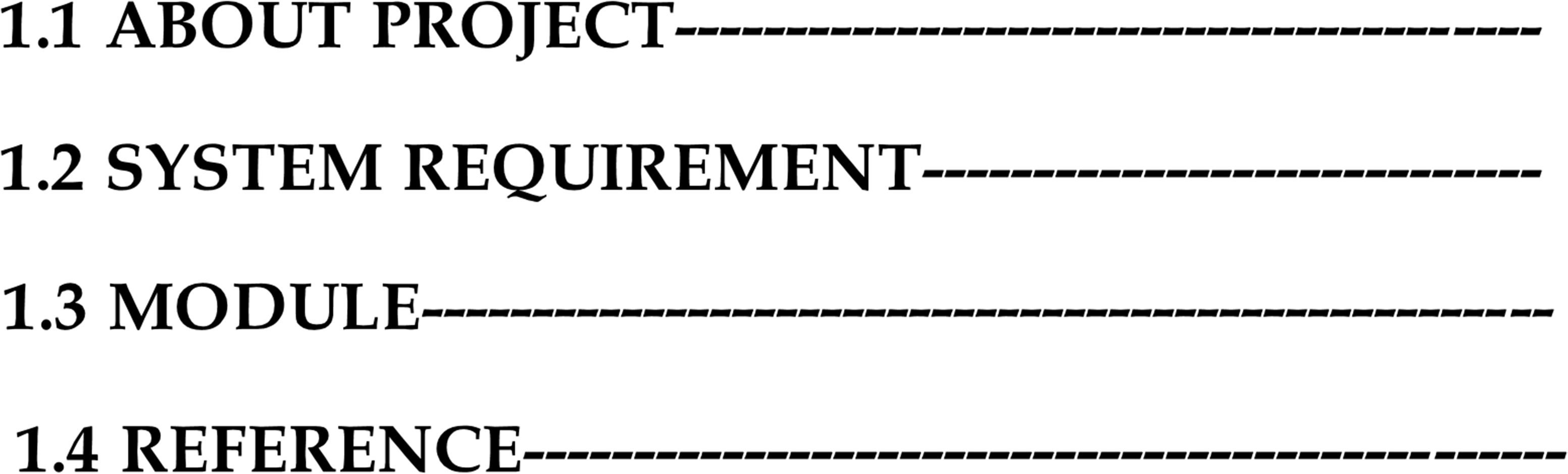
**SUBMITTED BY:** AYUSH PANDEY

**UNIVERSITY ROLL NO:** 2014607

**SECTION-** B

**SEMESTER** - 4

**SESSION-** 2020-2021



**1.1 ABOUT PROJECT:**

Today’s world is totally dependent on the use of internet. Internet has made communication easy. With the use of applications like what’s app, instagram, people are now able to connect with their friends and family.

The use of Email is the most dominating one when it comes to communicate with people officially. Companies rely on Emails to convey important messages to their employees rather than on what’s app and all. But what about the people who are visually impaired and can’t send and receive emails?

Voice Based Email is a program which helps people to send the email without actually typing it. This program also has a feature which will read the top most email in the inbox of a user. It will also tell the user about how many unread email you have not seen. This program may be beneficial for the people who are visually impaired. It may also benefit people who are suffering from spinal cord related problems and are unable to move. This project has been designed using python Programming Language and implements Google text to speech services.

this service will have the following features :

* ask for email login.
* Choice for voice instructor
* send messages without typing anything.
* inform total number of unread messages.
* read out the recent email.

**1.2 SYSTEM REQUIREMENT:**

1.2.1 Hardware Requirement:

 PROCESSOR USED: AMD RYZEN-5

 RAM: 1 GB (MINIMUM) USED- 4GB

 STORAGE: 99.5 MB

 OS USED: WINDOWS 10

1.2.2 Software used:

 PYCHARM-community 2021.1.1

 PYTHON 3.9.5

 Internet connectivity

**NOTE: TO MAKE THE PROJECT RUN USER MUST ALLOW THE PERMISSION TO SEND MAILS FROM LESS- SECURE PLATFORMS. THIS OPTION IS AVAILABLE SETTINGS OF YOUR ACCOUNT**

**1.3 DESCRIPTION OF PROJECT:**

 **PYTTSX3:**

It is a library which is to convert text to speech offline. It has many methods some of which have been used in the program.

1. Say (string):

It used to speak the string which is inside the parenthesis.

1. GetProperty():

used to get the properties of the speaking engine like voice (mail, female) , rate of speech, volume, id.

1. setProperty() :

used to set the properties like setting rate to 150,volume from 0.1 to 1, voice of male or female. Each word has to bepassed as an arguments are separated by a comma and its value.

1. RunAndWait ():

waits until say function is completed.

 **EmailMessage:**

This library is used to send email from the program. It import the structure from email message

 **speech \_Recognition:**

This package is used for speech recognition. Thus user will say something. It will Record that audio and convert the recorded text into a text and then the program will continue accordingly.

 **Imaplib:**

It is a library which is used to receive email from the program. We have to provide username and password in the variables and these library methods will log into the account and receive email.

 **server.login(username, password):**

login with the username and password provided earlier of the email whose message we want to receive.

 **Smtplib**:

This library is used to send email. Various functions of this library perform different steps to send an email to the desired person. First we set the host as “smtp.gmail.com”. Then we assign our username to from\_adrr variable, and senders email address to to\_adrrs variable. Similarly we assign message subject to the variables.

 **SMPT\_SSL (host, port):**

used to connect to the host services.

Eg- s.smtp(‘smtp.gmail.com’,587)

 **server.startttls():**

It informs to start the transport layer security. Several protocols use a command named STARTTTLS for this purpose. It is primarily intended as a countermeasure to passive monitoring. The example sends an email to a Mailtrap account with opportunistic TLS. The starttls() puts the connection to the SMTP serve**r** into TLS mode.

 **server.send\_message(email):**

used to send email from the address to sender in to the address in receiver .The message is stored is message. It is first converted to string and then sent.

 **win32com.client:**

This package contains a number of modules to provide access to automation objects. This package supports both late and early bindings, as we will discuss. Dispatch method when passed with the argument of **SAPI.SpVoice** It interacts with the Microsoft Speech SDK to speak what you type in from the keyboard.

 **get\_info():**

Function to get information from user end through microphone.it records the speech from the microphone and returns the text in lower order.

 **send\_email(receiver, subject, message):**

Function which is used to send email . it takes receiver as an reciever’s email id and also takes subject and message as an argument .it also cretes an SMTP server and then send message through server.send\_message(email).

 **login():**

Function to get information from user end about his email id and password and stores it in global variables.

 **unseen\_email():**

Function which counts the total number of emails in inbox .and also reads out the total number of unseen emails.

 **receive\_email():**

Function which reads out the latest unseen email from outlook Microsoft.

Messages are sorted according to time.

**Working:**

* **To send an email**  the user will be asked to speak send. The voice will be recorded and converted to text with the help of audio(“string”) function defined in input audio library

1. If string = normal : Audio is recorded for 4 seconds
2. If string = subject : Audio is recorded for 7 seconds
3. If string = message : Audio is recorded for 20 seconds

If he does so then he will be asked to speak the username of the person to whom he wish to send the email. A dictionary named **email\_list** has been used which contains username as key and their email addresses as value. As user speaks the username. The function **get\_info()** in input audio library will convert audio to text and will assign it to receiver variable the program will search the username in the dictionary and will give the corresponding email address. Then the email will be assigned to mail variable of smtplib library. The user then will be asked to speak the subject. The function **get\_info()** will convert audio to text and will assign it to Subject variable of smtplib library. In the end the user will be asked to speak message. The function **get\_info()** will convert audio to text and will assign it to message variable of smtplib library.

After this the **send\_email(receiver, subject, message):**define in smtplib will send the mail to the desired person. At last the function leaves the server and the mail would have been sent to the person and the program terminates.

**NOTE:**

server will depend from account to account.If your email address is of Gmail the server variable will be: **Server= smtplib.gmail.com** If your account is in outlook the server will be: **Server= smtp-mail.outlook.com**

* **To read the latest email** from inbox the user will be asked to speak check. inputaudio library **get\_info()** function will take input . The program logins to the server by **server.login(username, password)** then it selects inbox by server.select(“mail”). It then stores list of all mail in a list data by using server.search() method. Then we iterate over the list and separate all the mail ids in a list mail\_ids []. The last element of our list is our recent mail. We extract that and pass it to the fetch() function which extract all the details like senders mail, subject of mail and message and assign them to variable. After this the program speaks the details and it terminates. After taking input we used imaplib to read the latest email. The program first reads the senders email address then the subject of the mail and at last the message of the mail. After this the program terminates. If user speaks exit then the program terminates without doing anything. If user says something else the program gives appropriate message.

**REFERENCE:**

 Pyttsx3 library documentation

 Stack overflow

 Geeksforgeeks

 Google