



A PROJECT REPORT ON
“VOICE BASED EMAIL SENDING WITH GRAPHICAL USER INTERFACE”

SUBMITTED BY
ROSHAN SANJAY GHADGE
(ROLL NO.: 207758, EXAM SEAT NO. 207758)
T.Y.B.SC. I.T. - 2020-2021

UNDER THE GUIDANCE OF PROF. MRS. ASHWINI SOMNATHE

RAYAT SHIKSHAN SANTHA'S KARMVEER BHURAO PATIL
COLLEGE OF ARTS, COMMERCE AND SCIENCE [AUTONOMOUS]

DEPARTMENT OF INFORMATION TECHNOLOGY
JUHUNAGAR, SEC-15 A, NAVI MUMBAI-400703
DIST. THANE (M.S)



**Rayat Shikshan Sanstha's
KARMAVEER BHURAO PATIL COLLEGE, VASHI,
[Autonomous College]**



Reaccredited NAAC with Grade 'A+' (CGPA 3.53) | ISO 9001:2008 Certified
Institute 'Best College' Award by University of Mumbai

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This Is to Certify That

Mr./Miss: **ROSHAN SANJAY GHADGE**

Student of F. Y/S. Y/T. Y/ B.Sc.I.T., M.Sc.I.T., -Part(I/ II) Class From Karmaveer
Bhaurao Patil College, Vashi[Autonomous], Navi Mumbai Has Satisfactorily
Completed The Practical Course In Subject **PROJECT**

As per The Syllabus Laid by The University of Mumbai During the Academic Year
2020-2021

ROLL NO: **207758**

TEACHER

EXAMINER

HEAD

Date: 30/4/2021

Department Of Information Technology

ACKNOWLEDGEMENT

We have great pleasure in presenting the report on “Voice based email sending with GUI”.

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to express my special thanks to my project guide Miss Ashwini Somnath for their guidelines and support in completing my project.

I would also like to thankful to my colleagues who have helped me in successful completion of the projects.

INDEX

Sr. No.	Content	Page No.
1	Introduction	
	1. Introduction of the project	5
	2. Objective and Scope of Project	6
2	Analysis Phase	
	1. Requirement Gathering	7
	2. Cost Benefit Analysis	7
	3. Hardware evaluation factor	8
	4. Software evaluation factors	9
	5. Training Need	10
	6. Overview of technology used	11
3	Introduction of the Voice-Based Email Sending With GUI	13
4	Design Phase	
	1. Work Flow Diagram	16
5	Working	17
6	Screenshot	18
7	Coding Phase	23
8	Testing Phase	
	1. Unit Testing	33
	2. Integrating Testing	33
	3. System Testing	34
	4. Regression Testing	34
	5. Acceptance Testing	35
9	Maintenance and Evolution	36
10	Problem scope and system specification	37
11	Future scope of the project	37
12	Conclusion	38
13	References	39

INTRODUCTION OF THE PROJECT

Voice based email sending with graphical user interface project is developed using the python language. Here we are use some of the built-in python packages such as the Speech recognition, PyAudio, Smtplib, pyttsx3 and other for sending email with the voice based.

Firstly, we are use the python built in package is called tkinter with the help of the tkinter we are able to create the python graphical user interface i.e., Login panel window. Then here are taking the input form the user as username and password. User can login for the three time and if any new user is access for the first time then user needs to be sign up firstly.

Secondly, we are use the Speech recognition built-in python library which it will help to recognize the speech. Pyttsx3 is also one of the built-in python packages that it helps to convert the python text to speech. Also, there is python text to speech library is used which help for the text to speech. The information about the user is stored it into the database and at time of the login the username and password is match with the database record. If the data is match then user is able to login and if it is not match then user is not able to login it shows error.

New user is needs to be signup first the user needs to be entered their first name, middle name, last name, password, confirm password if the password and confirm password is match then it shows a message that password is match correctly otherwise it is going to show that the password is not match correctly. Once all information is fill up then user can go to login page and then log in.

Third, by using the SMTP (Send Email Transfer protocol) library we are firstly make the connection with it then we are using port 587 and valid email address and valid email password we are login to the Gmail server and then it will ask for the to, sub, content and at the end email will be send it to the email address.

RESEARCH OBJECTIVES

1. Project it will be also used into the organisation for sending the multiple of email and it is very easy to use.
2. It will also be useful for the students.
3. We can also use in the email marketing.
4. It will be mostly useful for blind people to access the multimedia functionality.

SCOPE OF PROJECTS

The idea of making the project is to while the technology is used into our every day of life. As we all are aware about the sending email to the anyone with the help the help of the graphical user interface. But in this project, we are trying to send an email with the help of the Voice-based system. Which is makes the very efficient to use and its time saving also. Here with using the different built-in python packages and library it helps for sending email with speech recognized and audio from the system and also, we can connect to the Gmail server and it is easy to send email to another person.

REQUIREMENT GATHERING

SOFTWARE

- Operating System: - **Windows OS**
- Front End Tool: Python IDE – **PyCharm 2020.3.3**
- Python Version: - **Python 3.0 or above**

COST BENEFIT ANALYSIS

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First, we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale. Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements. Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

HARDWARE EVALUATION FACTORS

When we evaluate computer hardware, we should first investigate specific physical and performance characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These hardware evaluation factors questions are summarized in the below figure. Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of ergonomics is also very important. Ergonomics is the science and technology that tries to ensure that computer and other technologies are "user-friendly", that is safe, comfortable and easy to use. Connectivity is another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

Hardware Evaluation Factors:

- Performance
- Cost
- Reliability
- Availability
- Compatibility
- Modularity
- Technology
- Ergonomics
- Connectivity
- Environmental requirements
- Software
- Support

SOFTWARE EVALUATION FACTORS

Software can be evaluated according to many factors similar to the hardware evaluation. Thus, the factors of performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support should be used to evaluate proposed software acquisitions. In addition, however, the software evaluation factors are summarized in below figure. For e.g., some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

SOFTWARE EVALUATION FACTORS:

- **EFFICIENCY:** Is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?
- **FLEXIBILITY:** Can it handle its processing assignments easily without major modifications?
- **SECURITY:** Does it provide control procedures for errors, malfunctions and improper use?
- **LANGUAGE:** Do our computer programmers and users write it in a programming language that is used?
- **DOCUMENTATION:** Is the software well documented? Does it include helpful user instructions?
- **HARDWARE:** Does existing hardware have the features required to best use this software?
- **PERFORMANCE:** Other characteristics of its performance, what about the cost, how much is reliable and etc.

TRAINING NEEDS

Training needs refer to the gaining of knowledge required for running the system. First of all, the system is a computer-based system therefore the person should have good knowledge about computer and it's working. He should know how to use software's on the computer. For a better usage and working of the software the organization should appoint a person who has good knowledge of all the required software. The organization gets a person trained through different institutes present in the market. The training should be as per the above requirements.

OVERVIEW OF TECHNOLOGY USED

Programming Language: Python

Features: -

Python is High-Level programming language which is developed by the Guido Van Rossum. The first version of python language is released into the 1991. Python language is named as Python after the BBC show “Monty Python’s Flying Circus”. Now a day’s python is getting more popular and it is strong community.

1. High level programming language
2. Python is free and open-source language
3. Python is simple
4. Error checking and high-level data type
5. Platform Independent language
6. User friendly
7. Extensible
8. Python is object oriented

Tool: Pycharm 2020.3.3

Features: -

Pycharm is dedicated Python **Integrated Development Environment** (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. This software it can run on the different of platform such as Windows, macOS, Linux.

1. Intelligent coding Assistance

2. Intelligent Code editor
3. Smart code navigator
4. Fast and Safe Refactoring
5. Debugging, Testing and Profiling
6. Database Tools

Operating System: Window OS

Features: - An operating system (OS) is software that act as an interface between the computer hardware components and the user. Every computer system is must have and at least one operating system to run the other program. Application like the Browsers, Ms office, Notepad, etc need some environment to run and perform its tasks.

1. Speed
2. Compatibility
3. Lower hardware requirement
4. Safety and security
5. Taskbar / Menu option

INTRODUCTION OF VOICE BASED EMAIL SENDING WITH GUI

What is Tkinter?

Tkinter is a python interface with the Tk GUI toolkit that is shipped with the python. Tkinter is standard GUI library for the creation of fast and easy GUI application. It provides a powerful object-oriented interface to the Tk GUI toolkit. This is python built in library which is provided with the all-python version.

To use this library, we just simply need to import it and we can use efficiently.

What is pyttsx3?

Pyttsx3 is the text-to-speech conversion library in the python. This library is work offline and is compatible with both Python2 and Python3 version. An applications invoke the pyttsx3 init()

Factory function to get a reference to pyttsx3. Engine instance. It is very easy to use tool which converts the entered text into speech.

This pyttsx3 module supports two voices first is female and the other is male which is provided by the “sapi5” for windows.

Installation: -

For the installation of pyttsx3 module command into the python terminal is: **pip install pyttsx3**

What is Speech Recognition?

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc.

Speech input using a microphone and Translation of speech to text:

Speech to text translation: - This is done with the help of google speech recognition this requires an active internet connection to work however, there are certain offline recognition system such as PacketSphinx, but have a very rigorous installation process that requires several dependencies. Google speech recognition is one of the easiest to use.

Installation: -

For the installation of Speech Recognition module command into the python terminal is: **pip install SpeechRecognition**

What is pipwin?

pipwin is the complementary tool for the pip on windows. pipwin installs unofficial python packages binaries for windows provided by the Christoph Gohlke. Before we running the pipwin command we need to refresh it and also checked for updating regularly.

Installation: -

For the installation of pipwin module command into the python terminal is: **pip install pipwin**

What is PyAudio?

PyAudio is set of python bindings for PortAudio, a cross platform c++ library interfacing with audio drives. Since the PyAudio is depends on the PortAudio we need to have is installed successfully. But in some of the python version this PyAudio is giving problem into the installation and working on the window system.

Installation: -

For the installation of PyAudio module command into the python terminal is: **pipwin install PyAudio**

What is SMTP?

Simple Mail Transfer Protocol (SMTP) is a protocol which handles sending e-mail and routing e-mail between mail server. Python is providing the Smtplib module, which defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP.

Parameters: -

- **Host:** This is the host running your SMTP server. You can specify IP address of the host or a domain name.
- **Port:** if you are providing host arguments, then you need to specify the port, where SMTP server is listening.
- **local_hostname:** if your SMTP server is running on your local machine, then you can specify just localhost as of this option.

Installation: -

For the installation of smtplib module command into the python terminal is: **pip install smtplib**

What is Pillow?

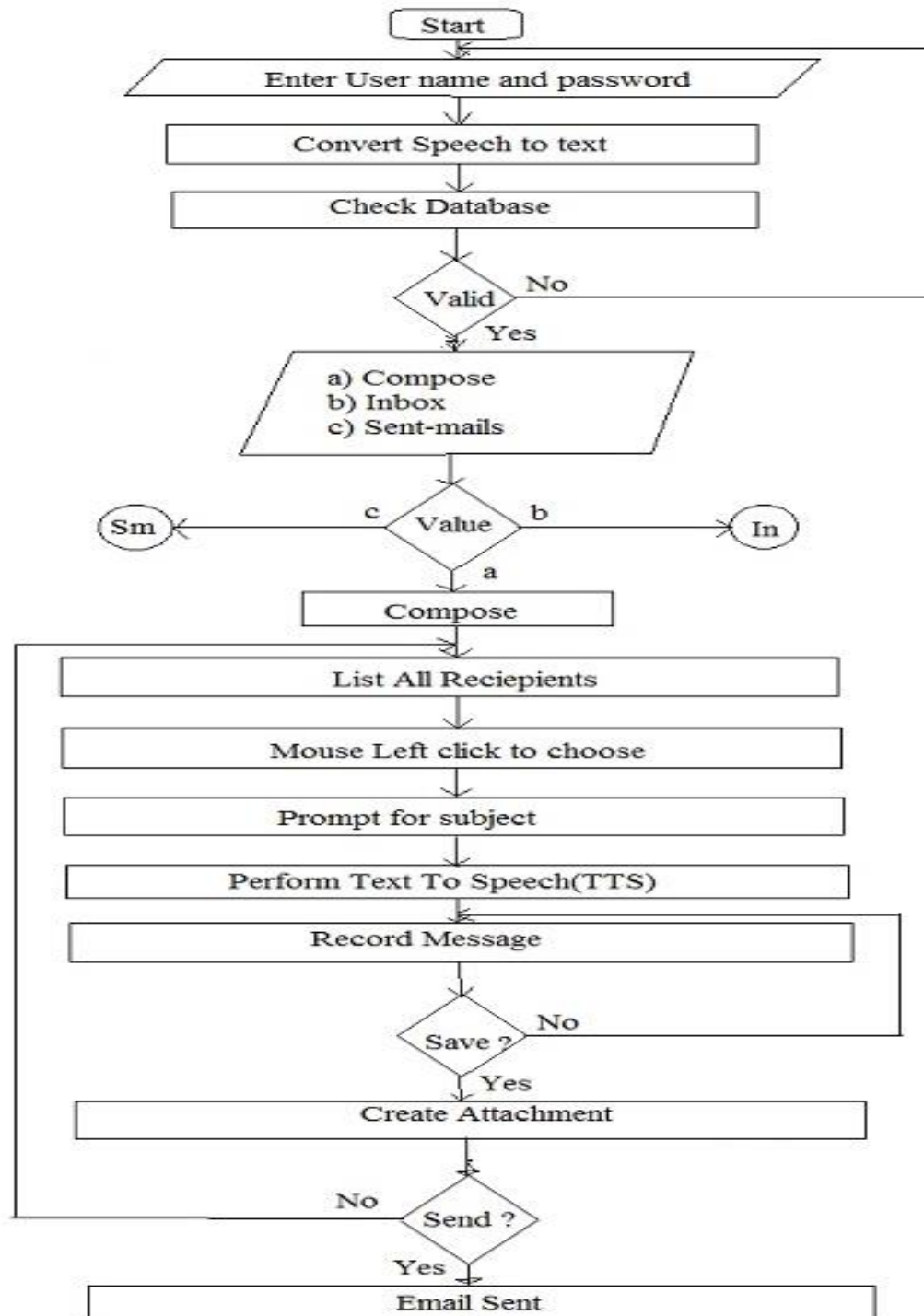
Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulation, and saving many different image file formats. It is available for windows, and other operating systems.

Pillow module is gives more functionalities runs on all major operating system and support for python 3. It supports wide range of images such as 'jpeg', 'png', 'bmp', 'gif', 'ppm', 'tiff' we can do almost all things into digital images using the pillow modules. Apart form the image processing functionality, including points operations, filtering images using built-in convolution kernels, and colours space conversion.

Installation: -

For the installation of pillow module command into the python terminal is: **pip install pillow**

FLOW DIAGRAM



WORKING

Voice-Based email sending system is capable of sending the email through the Gmail server using the Smtplib, and other some of python build in library. Here first the login window is show here user is need to enter their username and password. If new user is using for first time it needs to signup first. By entering the username and password these data is going to check with the available database of data if entered data is correct then user is log in. other it showing error likes (Invalid username, Invalid Password, All fields are required etc)

Once user is logged in then it will ask for to whom you want to send email, then what is subject for email, what is content in the email and at the last of it says that your email is send successfully.

Once the email is sent then new window is opened it showing that Graphics Interchange Format (GIF) image is shown. Which is shows that the mail is send successfully.

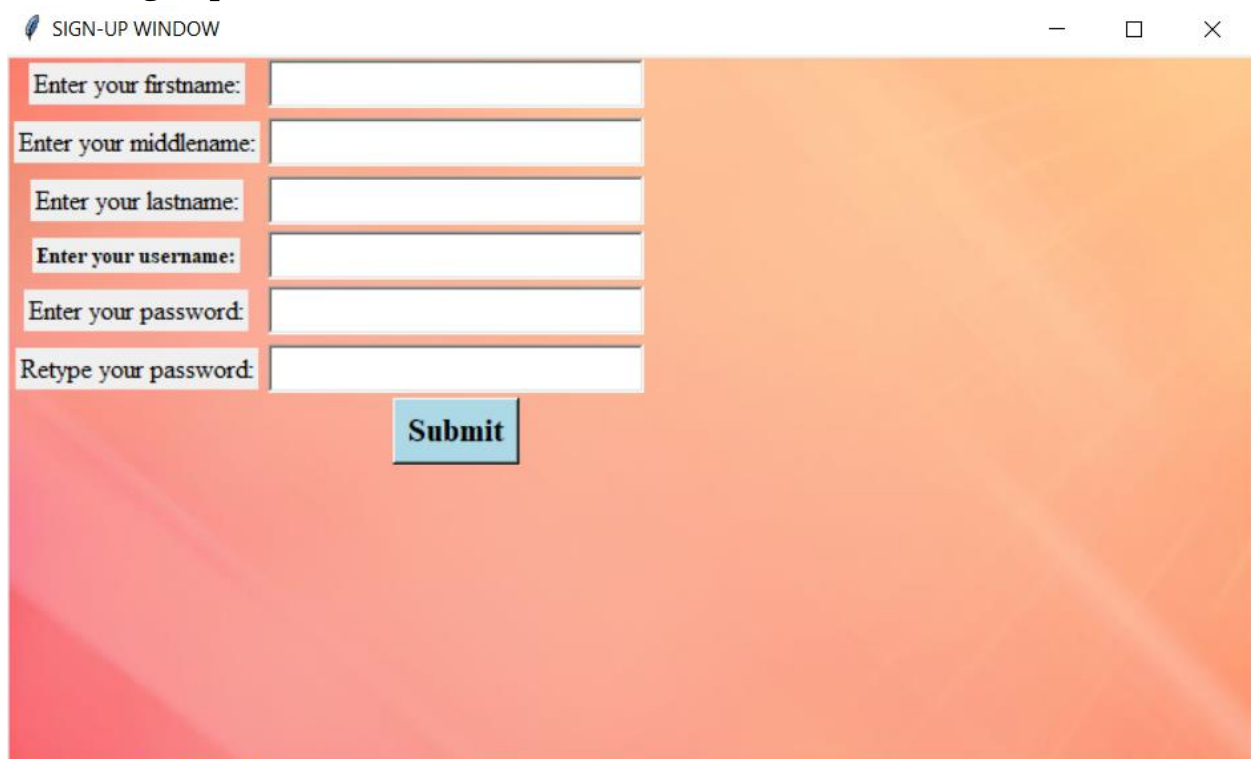
SCREENSHOTS

❖ Login Panel Window:




The screenshot shows a web browser window titled "Login System". The page has a yellow header bar with the text "Login System" in red. Below the header, there is a white rectangular area containing a circular profile picture of a man with dark hair, wearing a white shirt and a red tie. Below the profile picture, there are two input fields: "Username" and "password". The "password" field has a blue padlock icon to its left. Below the input fields, there are two buttons: "Login" and "Signup".

❖ Sign-up Panel Window:



The screenshot shows a web browser window titled "SIGN-UP WINDOW". The page has a white background with a light orange gradient on the right side. On the left, there are six input fields stacked vertically, each with a label to its left: "Enter your firstname:", "Enter your middlename:", "Enter your lastname:", "Enter your username:", "Enter your password:", and "Retype your password:". Below these input fields, there is a blue button labeled "Submit".

❖ Login with New User:

 — □ ×

Enter your firstname:	Harshal
Enter your middlename:	Subhash
Enter your lastname:	Ghadge
Enter your username:	harshal
Enter your password:	*****
Retype your password:	*****

Submit

PASSWORD IS MATCH SUCCESSFULLY
NEW REGISTRATION IS COMPLETED

(Once we fill all information and click on the submit button then message is show that password is match and new registration is get completed)

❖ Invalid Username: (If we enter wrong username)

 — □ ×

Login System


 **Username**
 **password**
Login **Signup**



Invalid user...
OK

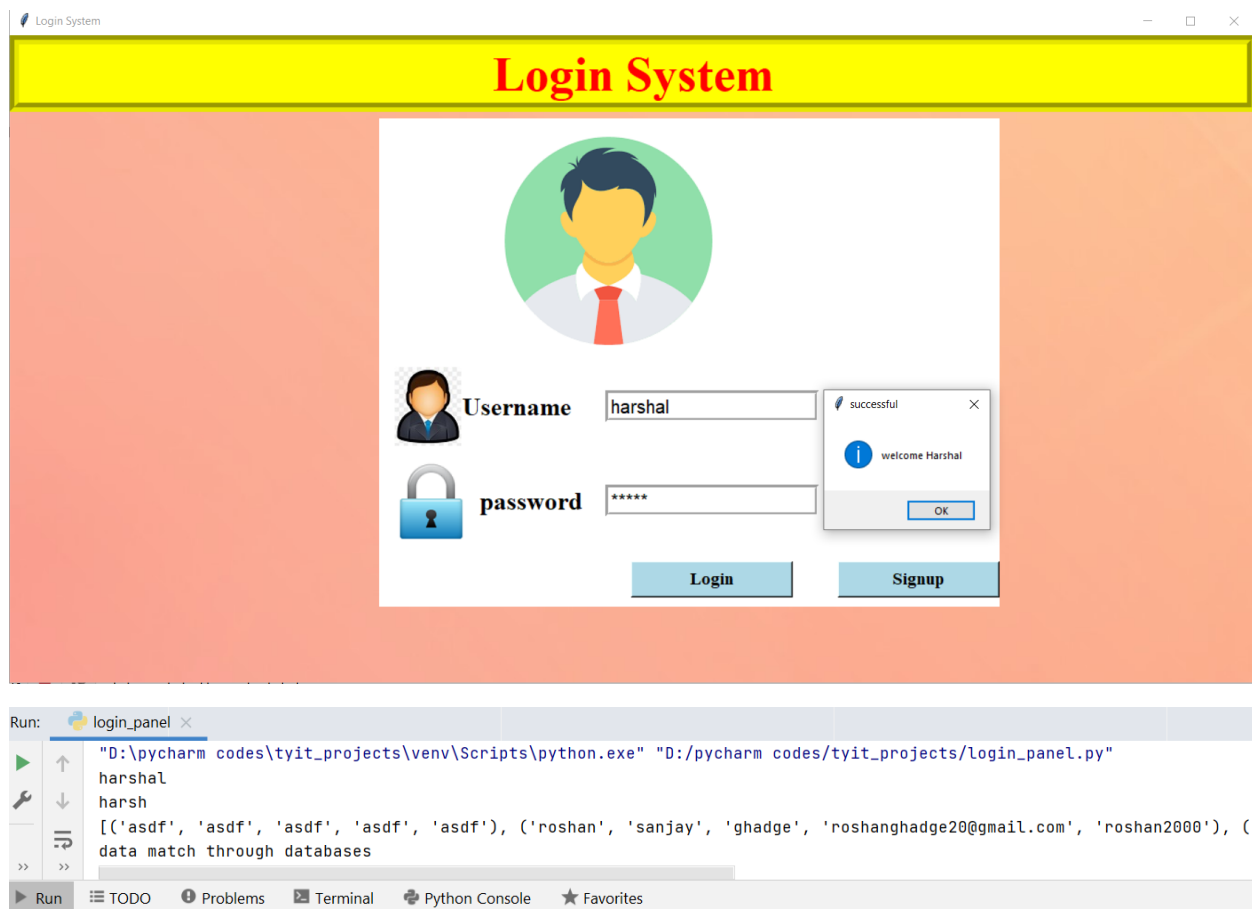
❖ **Invalid Password: (If we enter wrong password)**



❖ **All fields are required: (If user is directly click on the Login button)**



❖ Welcome Login:



(If username which is enter by user and database available data is match then only it show that welcome window)

❖ Email-Sending Window:



(Once login is successful it will ask for the to whom we want to send email, what is subject for email, what is content for email and at last it show that your email is send)

❖ **GIF Window:**



EMAIL SEND GIF



(Once Email is send successfully then this gif window is shown)

CODING PHASE

Login_Panel.py

```
from tkinter import *
import tkinter as tk
import PIL
from PIL import Image
from tkinter import messagebox
from PIL import ImageTk
from tkinter import PhotoImage
import mysql.connector
import Email_Send_gif as obj2
import voicebased_emailsend as obj3
mydatabase = mysql.connector.connect(host="localhost", user="root",
password="sshubhamg", port="3306", database="project")
cursor = mydatabase.cursor()
# query="create database project"
# query1="create table sample_project1(username varchar(30),password varchar(30));"
# query2='insert into sample_project1(username,password)
values("Roshan1","roshan@1234");'

# query="create table login(username varchar(30),password varchar(30));"
# query="create table signup(firstname varchar(30),middlename varchar(30),lastname
varchar(30),username varchar(30),password varchar(30));"

class Login_System:
    def __init__(self, root):
        self.root = root
        self.root.title("Login System")
        self.root.geometry("1350x700+0+0")
        self.bg_icon = ImageTk.PhotoImage(file="background .jpg")
```

```

self.user_icon = ImageTk.PhotoImage(file="man user.png")
self.pass_icon = ImageTk.PhotoImage(file="lock.png")
self.Logo_icon = PhotoImage(file="logo.png")
self.uname = StringVar()
self.pass_ = StringVar()
bg_lb1 = Label(self.root, image=self.bg_icon).pack()
title = Label(self.root, text="Login System", font=("times new roman", 40, "bold"),
bg="yellow", fg="red", bd=10, relief=GROOVE)
title.place(x=0, y=0, relwidth=1)
Login_Frame = Frame(self.root, bg="white")
Login_Frame.place(x=400, y=90)
logo1 = Label(Login_Frame, image=self.Logo_icon, bd=0).grid(row=0, columnspan=2,
pady=20)
lbluser = Label(Login_Frame, text="Username", image=self.user_icon,
compound=LEFT, font=("times new roman", 20, "bold"), bg="white").grid(row=1,
column=0, padx=1, pady=1)
txtuser = Entry(Login_Frame, bd=5, textvariable=self.uname, relief=GROOVE, font=("",
15)).grid(row=1, column=1, padx=20)
lblpass = Label(Login_Frame, text="password", image=self.pass_icon,
compound=LEFT, font=("times new roman", 20, "bold"), bg="white").grid(row=2,
column=0, padx=2, pady=1)
txtpass = Entry(Login_Frame, show="*", bd=5, relief=GROOVE, textvariable=self.pass_,
font=("", 15)).grid(row=2, column=1, padx=20)
btn_log = Button(Login_Frame, text="Login", command=self.login, width=15,
font=("times new roman", 14, "bold"), bg="light blue", fg="black").grid(row=3, column=1,
pady=10)
btn_log1 = Button(Login_Frame, text="Signup", command=self.Signup,
width=15, font=("times new roman", 14, "bold"), bg="light blue", fg="black").grid(row=3,
column=2, pady=10)

def login(self):

```



```

query2 = "select * from signup";
cursor.execute(query2)
user_info = cursor.fetchall()
check_username = self.uname.get()
print(check_username)
check_password = self.pass_.get()
print(check_password)
print(user_info)
if self.uname.get() == "" or self.pass_.get() == "":
    messagebox.showerror("Error", "All fields are required!!!")

for (firstname, middlename, lastname, username, password) in user_info:
    fst = firstname

    # query4='select * from login where username="check_username";'
    # cursor.execute(query4)

    if (username == check_username and password == check_password):
        print("data match through databases")
        messagebox.showinfo("successful", f"welcome {fst}")
        root.destroy()
        obj3.sendmail()
        obj2.Mail_send_gif()
        break

    elif (username != check_username and password == check_password):
        messagebox.showerror("Invalid username")
        break

    elif (username == check_username and password != check_password):
        messagebox.showerror("Invalid Password")
        break

```

```

def Signup(self):
    root.destroy()
    win = Tk()
    win.geometry("750x426")
    win.title("SIGNUP WINDOW")
    load = PIL.Image.open('bg1.jpg')
    render = ImageTk.PhotoImage(load)
    img = Label(win, image=render)
    img.place(x=0, y=0)
    lab1 = Label(win, text="Enter your firstname:", font=("times new roman", 12,))
    lab1.grid(row=0, column=1, padx=5, pady=5)
    fnameE = Entry(win, font=("", 15), bd=2)
    fnameE.grid(row=0, column=2)
    lab2 = Label(win, text="Enter your middlename:", font=("times new roman", 12,))
    lab2.grid(row=1, column=1, padx=5, pady=5)
    mnameE = Entry(win, font=("", 15), bd=2)
    mnameE.grid(row=1, column=2)

    lab3 = Label(win, text="Enter your lastname:", font=("times new roman", 12,))
    lab3.grid(row=2, column=1, padx=5, pady=5)
    lnameE = Entry(win, font=("", 15), bd=2)
    lnameE.grid(row=2, column=2)

    lab4 = Label(win, text="Enter your username:", font=("times new roman", 10,
"bold"))
    lab4.grid(row=3, column=1, padx=5, pady=5)
    usernameE = Entry(win, font=("", 15), bd=2)
    usernameE.grid(row=3, column=2)

    lab5 = Label(win, text="Enter your password:", font=("times new roman", 12,))
    lab5.grid(row=4, column=1, padx=5, pady=5)

```

```
passE = Entry(win, show="*", font=("", 15), bd=2)
passE.grid(row=4, column=2)
```

```
lab6 = Label(win, text="Retype your password:", font=("times new roman", 12,))
lab6.grid(row=5, column=1, padx=5, pady=5)
passrE = Entry(win, show="*", font=("", 15), bd=2)
passrE.grid(row=5, column=2)
```

```
def Submit():
```

```
    fname = fnameE.get()
    mname = mnameE.get()
    lname = lnameE.get()
    username1 = usernameE.get()
    password = passE.get()
    password1 = passrE.get()
```

```
if (password == password1):
```

```
    print("Password is match successfully")
```

```
    lab7 = Label(win, text="PASSWORD IS MATCH SUCCESSFULLY")
```

```
    lab7.grid(row=7, column=2)
```

```
    # messagebox.showinfo("NEW REGISTRATION IS SUCESSFUL")
```

```
    lab9 = Label(win, text="NEW REGISTRATION IS COMPLETED")
```

```
    lab9.grid(row=8, column=2)
```

```
else:
```

```
    print("Password is not match correctly")
```

```
    lab8 = Label(win, text="PASSWORD IS NOT MATCH CORRECTLY")
```

```
    lab8.grid(row=7, column=2)
```

```
    lab10 = Label(win, text="PLEASE ENTER VALID PASSWORD")
```

```
    lab10.grid(row=8, column=2)
```

```
    # messagebox.showinfo("PASSWORD IS NOT MATCH PLEASE ENTER VALID
    PASSWORD")
```

```

        query1 = 'insert into signup(firstname,middlename,lastname,username,password)
values("{}","{}","{}","{}","{}");'.format( fname, mname, lname, username1, password1)
        # query2='insert into login(username,password)
        values("{}","{}");'.format(username1,password1)

        print(fname)
        print(mname)
        print(lname)
        print(username1)
        cursor.execute(query1)
        mydatabase.commit()

        btn1 = Button(win, text="Submit", command=Submit, font=("times new roman", 15,
"bold"), fg="Black", bg="light blue")
        btn1.grid(row=6, column=2)
        win.mainloop()

root = Tk()
obj = Login_System(root)
root.mainloop()

```

voicebased_emailsend.py

```
import smtplib
import speech_recognition as sr
import pyttsx3
from email.message import EmailMessage

def sendmail():
    listener = sr.Recognizer()
    engine = pyttsx3.init()

def talk(text):
    engine.say(text)
    engine.runAndWait()

def get_info():
    try:
        with sr.Microphone() as source:
            print('listening...')
            voice = listener.listen(source)
            info = listener.recognize_google(voice)
            print(info)
            return info.lower()
    except:
        pass

def send_email(receiver, subject, message):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    # Make sure to give app access in your Google account
    server.login('shubhamghadge195353', 'XXXXXXXXXX')
```

```
email = EmailMessage()
email['From'] = 'shubhamghadge195353@gmail.com'
email['To'] = receiver
email['Subject'] = subject
email.set_content(message)
server.send_message(email)
```

```
email_list = {
    'roshan': 'roshanghadge20@gmail.com',
    'shubham': 'shubhamghadge559@gmail.com'
}
```

```
def get_email_info():
    talk('To Whom you want to send email')
    name = get_info()
    receiver = email_list[name]
    print(receiver)
    talk('What is the subject of your email?')
    subject = get_info()
    talk('what is the text in your email')
    message = get_info()
    send_email(receiver, subject, message)
    talk('Your email is sent')
    print('Your email is sent')
```

```
get_email_info()
```

```
#sendmail()
```

Email_send_gif.py

```
import tkinter as tk
import PIL
from PIL import ImageTk
from PIL import Image

def Mail_send_gif():
    look=tk.Tk()
    look.geometry("500x500")
    look.title("EMAIL_SEND_SUCCESSFULLY")
    img_file="email1.gif"
    info=PIL.Image.open(img_file)
    frames=info.n_frames # to calculate the total number os frames into image
    #print(frames) #to show the frames
    im=[tk.PhotoImage(file=img_file,format=f'gif -index {i}') for i in range(frames)]
    anim=None
    count=0
    def animation(count):
        global anim
        im2=im[count]
        gif_label.configure(image=im2)
        count+=1
        if count==frames:
            count=0
            anim=look.after(50,lambda :animation(count))

    gif_label=tk.Label(look,image='')
    gif_label.pack()
```

```
animation(count)
```

```
look.mainloop()
```


TESTING PHASE

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do. No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems

Levels of Testing

The different types of testing are as follows:

➤ Unit Testing:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

➤ Integration Testing:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

➤ **System Testing:**

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

- **Program(s) testing.**
- **String testing.**
- **System testing.**
- **System documentation.**
- **User acceptance testing.**

➤ **Regression Testing:**

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far, our project is concerned, we haven't tested our project using the field testing.

➤ **Acceptance Testing:**

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the book stores to check the system that whether the made system fulfils the goals or not.

MAINTENANCE AND EVALUTION

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First, we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale. Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements. Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

PROBLEM AND FUTURE SCOPE OF PROJECT

PROBLEMS: -

The following problem for this project was arrived at after reviewing the literature on voice based sending email with graphical user interface, and determining possible real-world situations where such systems would be of use. The following are problems were identified

1. This project is sending email with voice based so it is little bit of time consuming.
2. Here we are use Speech Recognition library of python for the recognition of what user is saying the accuracy for this module is 95% so it makes mistakes into some speeling.
3. For sending email with voice based it requires the constant of internet connection.

FUTURE SCOPE OF A PROJECT: -

The idea of making the project is to while the technology is used into our every day of life. As we all are aware about the sending email to the anyone with the help the help of the graphical user interface. But in this project, we are trying to send an email with the help of the voice-based system. Which is makes the very efficient to use and its time saving also. Here with using the different built-in python packages and library it helps for sending email with speech recognized and audio from the system and also, we can connect to the Gmail server and it is easy to send email to another person. In the Future we can also use some concepts of Artificial Intelligence or Machine Learning algorithms to use very efficiently.

CONCLUSION

The voice-based email sending with Graphical user interface is desktop application which is help to sending email with only voice-based system. This system it will incorporate all the features of the sending email (with connectivity to Gmail server, using port, checking email address, password). There is a database (MySQL) which it will store all the information about the login credential and new register of user also. At the time of the login the data is going to retrieve with the database and then it going to be check of validate. These all-retrieve data or the user enter data is display in the background. It is very easy to use and simple.

REFERENCES

- 1) Poonam Pate¹, Zeeshan Tamboli², Harsh Panchal³, Diksha Jain⁴ “Voice Based E-mail Application for, Blind/Visually Impaired People” IJARIE, 2017.
- 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 (IJETA), 2014.
- 3) <http://blog.jetbrains.com>
- 4) Ummuhan ysifa U., Nizar Banu P K “Voice Based Search Application “. In international journal of computational Engineering Research (IJCER).
- 5) Rranjal Ingle, Harshada Kanad, Arti Lanke “Voice based e-mail System for Blinds” International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 2016