

VOICE-ACTIVATED EMAIL ASSISTANT

**B R Srinivasa Rao¹, Harshini Sai², Mohammad Riyaz³, Mohammad Mukheed⁴,
Gottipadthala Nikhil⁵**

¹Assistant Professor, ACE Engineering College Hyderabad, India.

^{2,3,4,5}Student, ACE Engineering College Hyderabad, India.

Email srinivasa.bellary@gmail.com daughterofashoka@gmail.com riyazmohammad09876@gmail.com
21ag1a05g6@gmail.com nikhilgoud489@gmail.com

DOI: <https://www.doi.org/10.58257/IJPREMS38460> ABSTRACT

The Voice-Activated Email Assistant is a Python-based application that converts voice commands into text, enabling hands-free email composition and management. Designed for busy professionals and users with accessibility needs, it simplifies communication by supporting voice-activated file attachments and seamless email operations. Leveraging advanced speech recognition and natural language processing, the system ensures accurate transcription and improves efficiency, reducing the need for manual typing and saving time. With support for multiple languages and easy integration with email clients, this scalable tool enhances productivity while continually refining its performance through machine learning.

1. INTRODUCTION

The Voice-Activated Email Assistant is a Python-based application that enables users to manage their emails using voice commands. This system simplifies email-related tasks such as composing messages, attaching files, and sending emails by integrating voice recognition and text processing technologies. Designed for ease of use, it enhances productivity by reducing manual input and providing a hands-free solution for busy or accessibility-focused users.

2. OBJECTIVES

- Develop a system for composing and sending emails using voice commands.
- Integrate speech-to-text conversion to enhance user interaction.
- Facilitate the attachment of files through voice instructions.
- Ensure the system is user-friendly and efficient for daily use.

3. PROBLEM STATEMENT

Email management is a routine activity for most individuals, yet it requires significant manual effort. For users engaged in multitasking or those with physical impairments, typing and navigating traditional email platforms can be challenging. Existing tools do not fully cater to voice-controlled operations, creating a gap that this project aims to address by offering a comprehensive voice-activated solution.

4. PROPOSED SYSTEM

The Voice-Activated Email Assistant offers a solution by combining Python's speech recognition and automation libraries. Users can perform the following tasks:

- ❖ Compose emails by dictating the content.
- ❖ Attach files through voice commands.
- ❖ Send emails without manual typing or navigation.

The system processes voice input, converts it into text, and interacts with email servers to execute the required operations. This approach ensures accessibility and ease of use for diverse user groups.

5. SOFTWARE REQUIREMENTS

➤ **Programming Language:** Python

➤ **Libraries/Modules:**

- speech_recognition for voice input.
- smtplib for sending emails.
- pyttsx3 for text-to-speech feedback.
- os and email for file attachment handling.

Platform: Compatible with Windows, Linux, and macOS.

TECHNOLOGY DESCRIPTION:

□ **Speech Recognition:** Converts user voice commands into text using the speech_recognition library.

□ **Email Automation:** Use smtplib for secure communication with email servers to send messages and handle attachments.

□ **Text-to-Speech:** Employs pyttsx3 to provide audio feedback, ensuring users are informed of the system's actions.

ALGORITHM:

- Step 1: Start the program and initialize the microphone for input.
- Step 2: Capture voice input using the speech recognition module.
- Step 3: Convert the captured voice to text.
- Step 4: Parse the text to identify commands (e.g., compose, attach, send).
- Step 5: For composing emails:

- o Prompt the user for the recipient's email address.
- o Capture the email body via voice input.

Step 6: For file attachment:

- o Request the file name and location through voice.
- o Validate the file and attach it.

Step 7: Use smtplib to connect to the email server securely.

Step 8: Send the email and provide confirmation feedback via text-to-speech.

Step 9: Terminate the session or await further commands.

6. METHODS

1. Command-Line Interface (CLI)

- The system operates through a CLI, making it lightweight and easy to deploy across various platforms.
- The CLI prompts users for required inputs (e.g., recipient email, message body) and provides text-based feedback.
- Users interact with the program via voice commands, which are converted into actions by the system.

2. Speech Recognition

- The speech_recognition library was used to capture and process voice input.
- It converts voice commands into text, which is then parsed to identify specific actions like "compose," "attach file," or "send email."
- The library ensures high accuracy by using Google's speech-to-text API.

3. Email Automation

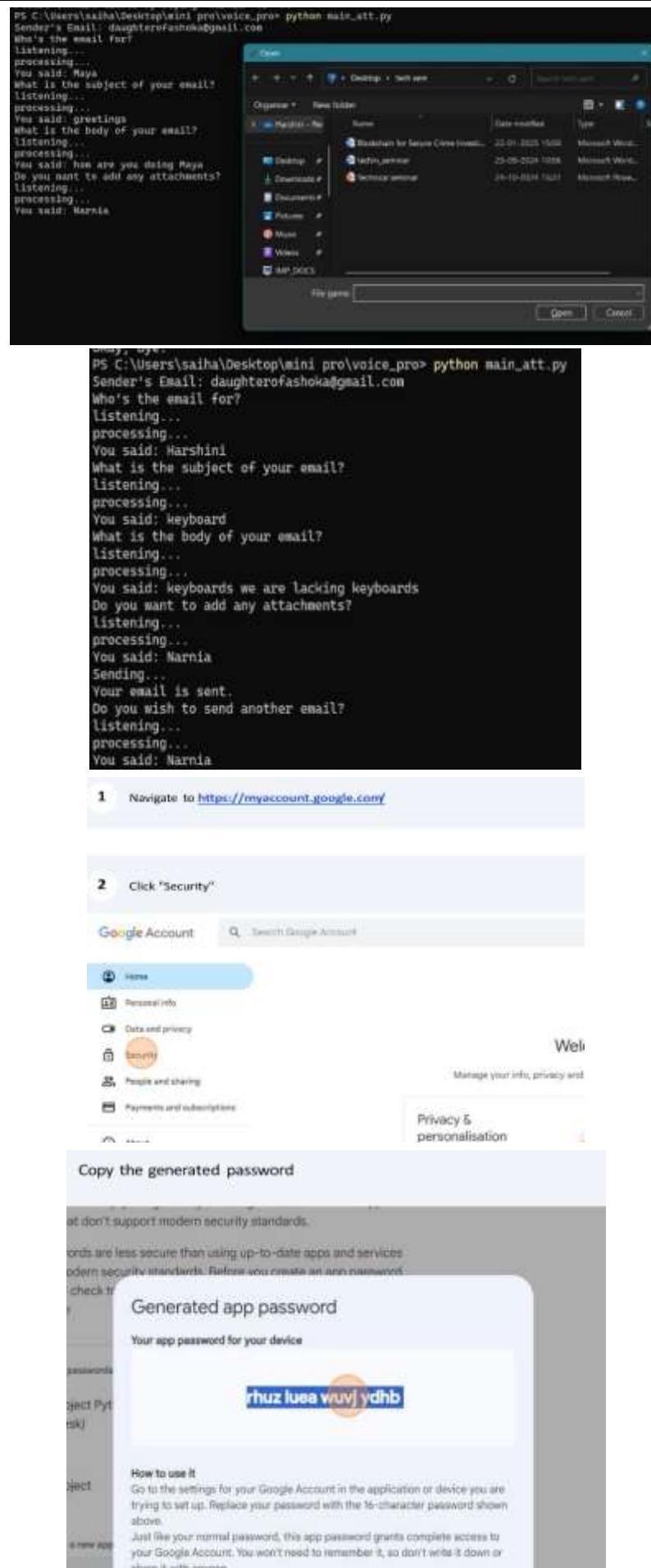
- The smtplib library is used to establish a secure connection with the email server (SMTP protocol).
- Email composition involves assembling the recipient address, subject, and body using the email module.
- File attachments are handled using MIME (Multipurpose Internet Mail Extensions) for proper formatting.

4. File Attachment Handling

- File attachments are managed through Python's os and email.mime libraries.
- The system validates file paths provided by the user and ensures that the files exist before attaching them to the email.

7. OUTPUT SCREEN

```
Who's the email for?  
listening...  
processing...  
You said: Harshini  
What is the subject of your email?  
listening...  
processing...  
You said: glass masks  
What is the body of your email?  
listening...  
processing...  
You said: computer keyboard  
Do you want to add any attachments?  
listening...  
processing...  
You said: no  
Sending...  
Your email is sent.  
Do you wish to send another email?  
listening...  
processing...  
You said: no  
Okay, Bye.  
PS C:\Users\saiha\Desktop\mini pro\voice_p
```



8. CONCLUSION

The Voice-Activated Email Assistant demonstrates the potential of voice-based automation in simplifying everyday tasks such as email management. By leveraging Python's robust libraries for speech recognition, email automation, and text-to-speech feedback, the system provides an efficient, hands-free solution for composing and sending emails. This project addresses accessibility challenges while enhancing user productivity, making it particularly useful for multitasking individuals or those with physical limitations. Future enhancements could include adding support for multiple languages, and integration with other email platforms for a more versatile application.

9. REFERENCES

- [1] https://en.wikipedia.org/wiki/Chrome_Web_Store
- [2] <https://gs.statcounter.com/browser-market-share>
- [3] https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/Web_Extensions/Browser_support_for_JavaScript_APIs
- [4] https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API#browser_support