CERTIFICATE

This is to certify that the project titled

"**Voice-Based Email & Messaging Assistant**"

(Hands Free Communication)

is a bonafide work carried out

by

Ms. Vedanti Lavekar,

MGM's College of Engineering, Nanded,

in partial fulfilment of the requirements for the

Infosys Springboard Virtual Internship Program,

December 2025 – June 2026.

This project has been developed under the guidance of the concerned faculty and industry mentors and represents the student's original work.

Dr. Santhiya Krishnasamy

Project Guide

Infosys Springboard Virtual Internship Program

ACKNOWLEDGEMENT

I express my sincere gratitude to Infosys Springboard for providing me with the opportunity to work on this meaningful and technically enriching project titled “Voice-Based Email & Messaging Assistant”. The structured learning modules, guidance, and resources provided through the internship program greatly contributed to the successful completion of this milestone.

I would like to thank my project guide **Dr. Santhiya Krishnasamy** for her continuous support, encouragement, and valuable guidance throughout the development of this project. Their suggestions and feedback helped me improve both the technical and documentation aspects of the work.

I am also thankful to Google Developers and open-source communities for providing powerful tools and APIs such as Gmail API, OpenCV, and OAuth 2.0, which played a crucial role in the implementation of this project.

With Reference**,**

Vedanti Nagesh Lavekar

TABLE OF CONTENTS

1.Introduction

1.1 Introduction

1.2 Aim & Objectives

1.3 Scope of the Project

1.4 Significance of the Project

1.5 Problem Statement

1.6 Literature Review

1.7 System Architecture

1.8 Technologies Used

1.8.1 Python

1.8.2 Flask Framework

1.8.3 HTML

1.8.4 CSS

1.8.5 JavaScript

1.8.6 Speech-to-Text Technology

1.8.7 OpenCV

1.8.8 LBPH Algorithm

1.8.9 Gmail API

1.8.10 Google OAuth 2.0

1.8.11 JSON

1.9 Project Outputs

Conclusion

ABSTRACT

Email communication has become an essential part of modern digital life; however, traditional email systems rely heavily on graphical interfaces, making them difficult to use for visually impaired users, elderly individuals, and people with physical disabilities. Additionally, manual email handling is inconvenient in hands-free situations such as multitasking or mobility-restricted environments.

This project, titled “Voice-Based Gmail Assistant with Face Authentication,” presents an intelligent and accessible solution that enables users to securely access and manage their Gmail accounts using voice commands and face recognition technology. The system integrates speech-to-text processing for voice interaction and computer vision techniques for biometric authentication, reducing dependency on keyboards and mouse devices.

The backend of the system is developed using Python and Flask, while OpenCV and the LBPH algorithm are used for face recognition. Gmail services are accessed securely using the Gmail API and Google OAuth 2.0, ensuring data privacy and secure authentication. The frontend is designed using HTML, CSS, and JavaScript to provide a user-friendly interface.

The proposed system improves accessibility, enhances security, and demonstrates the practical application of Artificial Intelligence and web technologies. This milestone-based implementation validates the feasibility of combining voice interaction and biometric authentication for secure email communication.

Infosys Voice-Based Email & Messaging Assistant – Hands-Free Communication

1.1 Introduction

Electronic mail has become one of the most essential communication tools in the modern digital era. It is widely used for personal communication, official correspondence, educational purposes, and business transactions. Among the various email service providers available today, Gmail is one of the most popular platforms due to its reliability, advanced security features, large storage capacity, and seamless integration with other Google services. Millions of users across the globe rely on Gmail for their day-to-day communication needs.

Despite its widespread adoption, Gmail and similar email platforms are primarily designed for users who can easily interact with graphical user interfaces using keyboards, mouse devices, and visual displays. Users are required to read text from screens, type responses manually, and navigate through multiple menus. This traditional mode of interaction creates significant challenges for certain categories of users, especially visually impaired individuals, elderly people, and users with physical disabilities. Such users may find it difficult or even impossible to use conventional email systems effectively.

In addition to accessibility issues, modern users often face time constraints and multitasking demands. Manually handling emails can be inconvenient while driving, cooking, or performing other tasks. These limitations highlight the need for alternative methods of interaction that allow users to access email services in a hands-free and efficient manner.

Recent advancements in artificial intelligence, speech recognition, and computer vision have opened new possibilities for building intelligent systems that enable users to interact with applications using voice commands and biometric authentication. Voice-based systems allow users to perform tasks without physical interaction, while face recognition provides a secure and convenient authentication mechanism.

This project, titled “Voice-Based Gmail Assistant with Face Authentication,” aims to address these challenges by developing an intelligent system that enables users to securely access and manage their Gmail account using voice commands and face recognition technology. The

system focuses on improving accessibility, usability, and security by combining modern web technologies with artificial intelligence techniques.

1.2 Aim & Objectives

The primary aim of this project is to design and develop a secure, accessible, and user-friendly voice-based Gmail assistant that enables users to authenticate using face recognition and interact with their Gmail account through voice commands.

The project aims to reduce the dependency on traditional input devices such as keyboards and mouse by providing a hands-free interaction mechanism. This approach is particularly beneficial for visually impaired users, elderly individuals, and users with physical limitations who may face difficulties using conventional email systems. By enabling voice-based interaction, the system enhances usability and promotes inclusivity.

Another important aim of this project is to enhance security by integrating biometric authentication in the form of face recognition. Traditional password-based authentication systems are vulnerable to security threats such as phishing attacks, password leakage, and unauthorized access. Face recognition provides an additional layer of security by ensuring that only authorized users can access the system.

The project also aims to demonstrate the practical application of modern technologies such as speech recognition, computer vision, and cloud-based APIs. Through this project, the developer gains hands-on experience in building real-world applications that combine multiple technologies to solve practical problems.

The objectives of the project define the specific goals that the system intends to achieve. These objectives guide the design and implementation of the system and ensure that the project meets its intended purpose.

One of the primary objectives of the project is to develop a voice-controlled Gmail assistant that allows users to access their email accounts securely. The system should support face recognition-based authentication to ensure that only authorized users can log in. Another objective is to integrate Gmail services using the official Gmail API, allowing users to access their inbox securely without storing sensitive credentials.

Additional objectives include improving accessibility for visually impaired and elderly users by enabling voice-based email interaction. The project also aims to enhance system security by

implementing multiple authentication methods such as password login, Google OAuth login, and face recognition.

Furthermore, the project seeks to provide practical exposure to technologies such as OpenCV for face recognition, speech-to-text systems for voice commands, and Flask for backend development. By achieving these objectives, the project demonstrates the effective integration of artificial intelligence and web technologies.

1.3 Scope of the Project

The scope of the project defines the boundaries and functionalities covered in Milestone-1. The project focuses on developing a prototype system that demonstrates the feasibility of integrating voice interaction and face authentication with Gmail services.

The scope includes user registration, face image capture, and face recognition-based login. The system supports basic voice commands that allow users to interact with their Gmail inbox. Secure session handling and backend processing are also included within the scope of the project.

However, the project has certain limitations. The system supports only a limited set of voice commands and basic Gmail operations. Advanced features such as composing emails using voice, offline speech recognition, and large-scale deployment are outside the scope of Milestone-1. These limitations are considered acceptable for a milestone-based academic project.

1.4 Significance of the Project

The significance of this project lies in its ability to address real-world challenges related to accessibility, security, and usability in email systems. By enabling voice-based interaction, the system improves accessibility for users who find traditional email interfaces difficult to use.

The integration of face recognition enhances security by reducing reliance on passwords, which are often vulnerable to security threats. Biometric authentication provides a more secure and convenient alternative to traditional authentication methods.

From an academic perspective, the project is significant as it demonstrates the practical application of artificial intelligence, computer vision, and speech recognition technologies. It also provides valuable experience in API integration and web application development, making it an important learning exercise.

1.5 Problem Statement

In today's digital world, email is one of the most essential means of communication for personal, educational, and professional purposes. However, most existing email systems, including Gmail, are designed to be operated using traditional graphical interfaces that require manual typing and visual interaction. This makes email access difficult for visually impaired users, elderly individuals, and people with physical disabilities. Even for regular users, manually handling emails can be inconvenient while multitasking or in hands-free situations.

In addition to accessibility issues, traditional password-based authentication systems pose security risks such as password theft, phishing attacks, and unauthorized access. Remembering complex passwords is also challenging for many users, often leading to weak security practices.

Therefore, there is a need for a secure and accessible email system that allows users to interact with their Gmail account using voice commands while ensuring strong authentication through advanced methods such as face recognition. The system should improve usability, enhance security, and provide a hands-free email experience.

Problem Statement: To develop a secure and accessible voice-based Gmail assistant that enables hands-free email interaction using voice commands and face recognition authentication.

1.6 Literature Review

Several studies and existing systems have explored voice-based assistants and biometric authentication independently. Popular voice assistants such as Google Assistant and Amazon Alexa provide voice-based interaction but are not specifically designed for secure Gmail access using face recognition.

Fig 1.6 Voice Assiatant Icon

Research in the field of face recognition has shown that algorithms such as Local Binary Pattern Histogram (LBPH) are effective for real-time applications with small datasets. OpenCV provides efficient tools for face detection and recognition using Haar Cascade classifiers.

Previous works on Gmail API integration highlight the importance of OAuth 2.0 for secure authentication and authorization. However, integrating voice interaction, face recognition, and Gmail services into a single unified system remains a complex challenge. This project aims to bridge this gap by combining these technologies into a single platform.

1.7 System Architecture

The system follows a client-server architecture where the frontend acts as the user interface and the backend handles application logic and data processing. The frontend is developed using HTML, CSS, and JavaScript, allowing users to interact with the system through a web browser.

The backend is implemented using Flask, which manages authentication, face recognition processing, voice command handling, and communication with Gmail APIs. External services such as Gmail API and speech recognition services are integrated through secure APIs.

This architecture ensures modularity, scalability, and secure communication between different components of the system.

1.8 Technologies Used

The successful development of the Voice-Based Gmail Assistant with Face Authentication requires the integration of multiple technologies from different domains such as web development, artificial intelligence, computer vision, speech processing, and cloud services. Each technology used in this project has been carefully selected based on simplicity, feasibility, performance, and suitability for an academic milestone-based project. The following section provides a detailed explanation of all technologies used in the implementation.

1.8.1 Python (Core Programming Language)

Python is used as the core programming language for the development of this project. Python is a high-level, interpreted programming language known for its simplicity, readability, and extensive library support. It is widely used in the fields of artificial intelligence, machine learning, computer vision, and automation, making it an ideal choice for this project.

In this project, Python is used to implement the backend logic, handle user authentication, process face recognition, integrate Gmail APIs, and manage data storage. Python provides built-in modules such as os, json, hashlib, and base64, which are used for file handling, data storage, password hashing, and image encoding respectively. The availability of third-party libraries such as OpenCV and Google API clients further enhances Python's suitability for this application.

The use of Python allows rapid development and easy debugging, which is essential for milestone-based academic projects. Its platform independence ensures that the project can run on different operating systems without major modifications.

1.8.2 Flask Framework (Backend Development)

Flask is a lightweight web framework written in Python and is used for backend development in this project. Flask follows a micro-framework approach, meaning it provides only the essential tools required to build web applications, allowing developers to add additional features as needed.

In this project, Flask is responsible for handling HTTP requests and responses, defining application routes, managing user sessions, and connecting the frontend with backend services. Flask routes are used to implement APIs for login, registration, face authentication, Gmail access, and voice command processing.

Flask also provides session management capabilities, which are used to track authenticated users and restrict access to authorized pages such as the dashboard. Due to its simplicity and flexibility, Flask is widely used in academic projects and prototypes where rapid development and clarity of code are important.

1.8.3 HTML (HyperText Markup Language)

HTML is used to create the structure of the web pages in the application. It defines the layout and elements such as input fields, buttons, headings, and containers used in the login, registration, and dashboard pages.

In this project, HTML provides the basic skeleton of the user interface. Forms are created using HTML to collect user inputs such as email, password, and button interactions. Video and canvas elements are also included using HTML to support camera access for face capture during registration and login.

HTML ensures that the application is accessible through any modern web browser, making the system platform-independent.

1.8.4 CSS (Cascading Style Sheets)

CSS is used to design and style the user interface of the application. It controls the visual appearance of web pages, including layout, colors, fonts, spacing, and responsiveness.

In this project, CSS is used to create a clean, professional, and user-friendly interface. Styling improves readability and usability, especially for visually impaired users. Buttons, forms, and containers are designed to be visually distinct and easy to interact with. CSS also helps maintain consistency across different pages of the application, ensuring a uniform look and feel. A well-designed interface enhances user experience and makes the application more intuitive.

1.8.5 JavaScript (Client-Side Scripting)

JavaScript is used as the client-side scripting language to handle dynamic interactions within the application. It enables real-time communication between the frontend and backend using asynchronous API calls.

In this project, JavaScript is used to:

- Capture voice input from the microphone
- Access the camera using browser media APIs
- Capture and send face images to the backend
- Call backend APIs using the fetch () method
- Display success and error messages dynamically

JavaScript plays a crucial role in implementing speech-to-text functionality and face login features. By handling tasks on the client side, JavaScript reduces server load and improves system responsiveness.

1.8.6 Speech-to-Text Technology (Google Browser-Based Speech Recognition)

Speech-to-text conversion is an essential component of the voice-based Gmail assistant. In this project, Google's browser-based speech recognition is used for converting spoken voice commands into text.

This technology allows the system to capture voice input through the microphone and convert it into textual commands in real time. These commands are then sent to the backend for processing. Google speech recognition is chosen because it offers good accuracy, fast response time, and easy integration with JavaScript.

The use of browser-based speech recognition eliminates the need for heavy model downloads or complex setup, making it suitable for Milestone-1 implementation. Whisper is mentioned as a future enhancement for offline and advanced speech recognition but is not used in the current implementation.

1.8.7 OpenCV (Computer Vision Library)

OpenCV (Open Source Computer Vision Library) is used for implementing face detection and face recognition in the project. OpenCV provides powerful tools and pre-trained models for image processing and computer vision tasks.

In this project, OpenCV is used to detect human faces from images captured through the camera. Haar Cascade classifiers provided by OpenCV are used for face detection. Once a face is detected, it is processed further for recognition.

OpenCV is widely used in academic and industrial applications due to its efficiency, accuracy, and open-source nature. Its compatibility with Python makes it easy to integrate with the backend system.

1.8.8 LBPH Algorithm (Local Binary Pattern Histogram)

The Local Binary Pattern Histogram (LBPH) algorithm is used for face recognition in this project. LBPH is a traditional and efficient face recognition algorithm that works well with small datasets and varying lighting conditions.

The algorithm works by analyzing the local texture of the face image and converting it into binary patterns. These patterns are then represented as histograms, which are used to identify and compare faces.

LBPH is chosen because it is simple to implement, computationally efficient, and suitable for real-time applications. It does not require large datasets or high-end hardware, making it ideal for academic projects.

1.8.9 Gmail API (Email Integration)

The Gmail API is used in this project to provide secure and programmatic access to Gmail services. It allows the application to interact with a user's Gmail account and perform operations such as reading inbox emails in a structured and controlled manner. By using the Gmail API, the application can retrieve email data such as sender details, subject lines, and message content without directly accessing or storing the user's Gmail password.

The Gmail API follows Google's security guidelines and ensures that only authorized applications can access user data. It provides a standardized way to communicate with Gmail servers using HTTPS requests, making email access reliable and secure. In this project, the Gmail API helps automate email-related tasks and enables integration of Gmail functionality with voice commands, thereby improving usability and accessibility.

1.8.10 Google OAuth 2.0 (Authentication)

Google OAuth 2.0 is used in this project for secure authentication and authorization. OAuth 2.0 is an industry-standard protocol that allows users to log in using their Google account without sharing their actual password with the application. Instead of storing passwords, OAuth generates secure access tokens that grant limited permission to access Gmail services.

When a user chooses to log in using Google, they are redirected to Google's authentication page, where they grant permission to the application. After successful authentication, Google provides an access token that is used by the application to communicate with the Gmail API. This approach ensures that sensitive user credentials are never stored within the application, significantly reducing security risks.

By using Google OAuth 2.0, the project follows best security practices and provides a safe and trusted login mechanism, making the system more secure and reliable.

1.8.11 JSON (Data Storage)

JSON (JavaScript Object Notation) is a lightweight data-interchange format that is used in this project for storing and managing application data. It is easy to read, write, and parse, making it highly suitable for small-scale and prototype-based applications such as this milestone project.

In this project, JSON is used to store user-related information and face recognition metadata. Files such as users.json are used to store registered user credentials in a structured format, where passwords are stored in encrypted (hashed) form for security purposes. Another JSON

file, labels.json, is used to store the mapping between face recognition labels and user identities. This mapping helps the system identify which registered user corresponds to a recognized face during face login.

JSON is chosen over traditional databases because it provides simplicity and ease of implementation. Since Milestone-1 focuses on core functionality rather than large-scale deployment, JSON storage is sufficient and efficient. It eliminates the need for database configuration while still allowing structured data handling.

Additionally, JSON integrates seamlessly with Python, as Python provides built-in libraries to read and write JSON data easily. This allows smooth communication between different components of the system, such as authentication, face recognition, and session management.

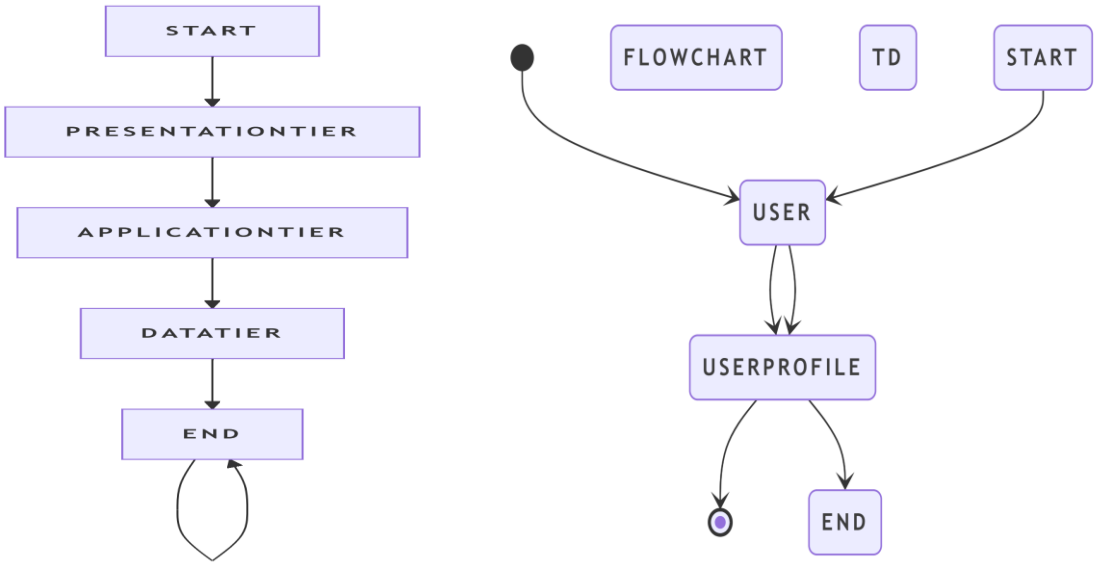
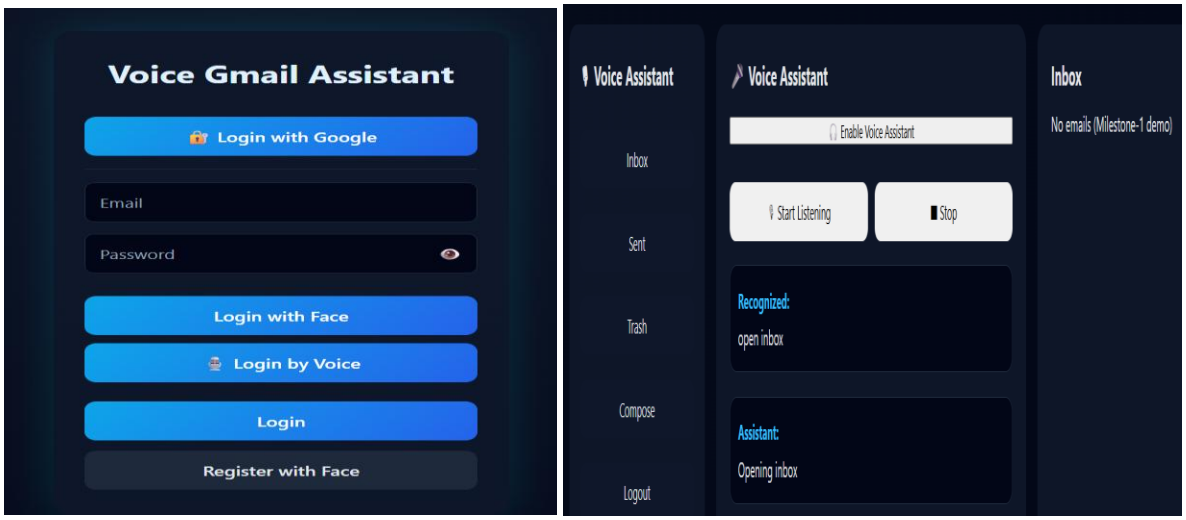


Fig 1.8 Frontend and Backend Workflow



1.9 Project Outputs

CONCLUSION

The Voice-Enabled Email Assistant project successfully demonstrates the design and implementation of an accessible, secure, and user-friendly email interaction system. The primary objective of this project was to assist visually impaired users by enabling hands-free email access through voice commands, thereby reducing dependency on traditional graphical interfaces. The system integrates voice-based interaction with multiple authentication mechanisms to ensure both usability and security.

In this project, a web-based architecture was implemented using Python Flask as the backend framework and HTML, CSS, and JavaScript for the frontend interface. Voice interaction was achieved using browser-based Speech-to-Text and Text-to-Speech technologies, allowing users to issue commands and receive audible feedback. Authentication mechanisms such as manual login, Google OAuth 2.0, and face recognition were incorporated to provide flexible and secure access options. Sensitive user information is protected through password hashing and token-based authentication, ensuring data security and privacy.

The project meets all the objectives defined for Milestone-1, including successful user authentication, voice command recognition, voice-based navigation, and accessibility support. Although the current implementation demonstrates limited email functionality, it establishes a strong foundation for future enhancements. Additional features such as full email composition, advanced inbox management, and improved voice intelligence can be integrated in subsequent milestones.

Overall, this project highlights the effective use of voice technologies and secure authentication methods to create an inclusive digital solution. It demonstrates practical application of modern web technologies and provides a scalable framework that can be further extended to support real-world deployment and advanced assistive features.