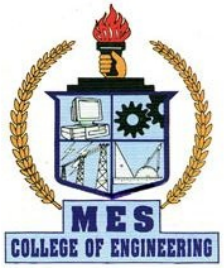
Voice Based Email For Visually Challenged

A Main Project Report
submitted in partial fulfillment of the requirements for the award of the Degree of

Master Of Computer Application
UNDER
APJ Abdul Kalam Technological University

BY

Jeniya Majeed T
MES21MCA-2020



DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING
KUTTIPPURAM, MALAPPURAM - 679 582

May 2023

Declaration

I undersigned hereby declare that the main project report **Voice Based Email For Visually Challenged** submitted in partial fulfillment of the requirements for the award of *Degree of Master of Computer Application* under *APJ Abdul Kalam Technological University* is a bona fide record done by me under the supervision of **Prof. Balachandran K P**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma, or similar title of any other University.

Sincerely,
Jeniya Majeed T
MES21MCA-2020

Place: Kuttippuram

Date:

MES COLLEGE OF ENGINEERING

KUTTIPPURAM, KERALA - 679 582
(A NAAC ACCREDITED INSTITUTION WITH NBA ACCREDITED DEPARTMENTS, APPROVED BY AICTE AND AFFILIATED TO APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY)

DEPARTMENT OF COMPUTER APPLICATION

CERTIFICATE

This is to certify that the main project report entitled **Voice Based Email For Visually Challenged** is a bona fide record of the main project work carried out by **Jeniya Majeed T (Register No: MES21MCA-2020)**, fourth-semester student of the department, during the academic year 2022-23, in partial fulfillment of the requirements for the award of *Degree of Master of Computer Application* under *APJ Abdul Kalam Technological University*. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

External Examiner Head of the Department

Acknowledgement

My endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my main project.

I pay my gratitude to the Almighty for His invisible help and blessing for the fulfillment of this work.

At the outset I express my heart full thanks to our Head of the Department, **Prof.Hyderali. K** for permitting me to do this main project.

I would like to express my sincere gratitude to, our Main Project coordinator, **Mr. Vasudevan T V**, Assistant Professor in the Department of Computer Application for his exceptional support and encouragement throughout this project.

I take this opportunity to express my profound gratitude to my guide, **Prof. Balachandran K P**, Associate Professor in the Department of Computer Application for his valuable guidance.

I am also grateful to all our teaching and non-teaching staff for their encouragement, guidance, and wholehearted support.

Last but not least, I am gratefully indebted to my family and friends, who gave me their precious help in my main project.

Sincerely,
Jeniya Majeed T
MES21MCA-2020

Abstract

The Internet is considered a major amenity in today's world. However, not all people can use the internet as the user need to know what is written on the screen. This makes the internet a completely useless technology for visually impaired people. E-mails are the most dependable way of communication over the Internet for sending and receiving information. As nearly 285 million people worldwide are estimated to be visually impaired and it is necessary to provide internet facilities for them. Therefore this project develops a voice-based email system that will aid visually impaired people to use the services for communication.

The main purpose of this project is to guide visually challenged users. The proposed system focuses on providing basic functionalities like composing, reading, sending, and receiving emails along with voice-based assistants. As the input to the system does not use a keyboard or mouse, users can easily give input by speaking the message. Thus, the system proposed is entirely different from the existing ones. The user will be able to give commands to the system which the system will follow.

The existing mail services do not provide easy access to visually challenged people because there are no readout options to hear the mail that is received at their mail addresses. In this proposed system, the speech is read in a regional language (Malayalam) to avail understandability of blind people. This system is completely based on interactive voice response which will make it user-friendly and efficient to use. Here, we use Python language to implement speech-to-text and text-to-speech.

This application can help in overcoming some of the drawbacks of the existing email systems. In this system, the use of the keyboard has been eliminated completely. The user only requires listening to the voice commands given by the system and responding accordingly in order to perform the desired operations. It aims to help visually impaired people to be a part of the growing digital India by using the internet and also to make the life of such people quite easily. The success of this project will also encourage developers to build something more useful for visually impaired or illiterate people, who also deserve an equal standard in society.

Table Of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Voice-Based Email project is designed to address the communication challenges faced by visually challenged individuals while accessing and managing their email accounts. The project aims to provide a solution that enables visually challenged individuals to compose, send, and manage their emails using their voices1. The project utilizes the SpeechRecognition library in Python to convert the user's voice into text and send emails using the smtplib library.

This project has several key features, including voice input, text-to-speech, and speech-to-text functionalities and also partially converts to a regional language (Malayalam). The project allows visually challenged individuals to compose emails using their voice and transcribes the received emails into speech for the user to listen to. The project also enables the user to send emails using their voice by transcribing the user's voice into text. Additionally, the project includes accessibility features, such as user-friendly interfaces and error-handling mechanisms.

The Voice-Based Email project provides several benefits to visually challenged individuals, such as increased independence in accessing and managing their email accounts. The project allows visually challenged individuals to compose and send emails with ease, improving their communication abilities. The project's text-to-speech feature makes it easier for visually challenged individuals to listen to their emails, enhancing their user experience and making the email communication process more accessible and inclusive.

1.1 Background

Visually challenged individuals face significant challenges in accessing and managing their email accounts, particularly when it comes to composing, sending, and managing emails. Traditional email clients and interfaces are not designed with accessibility features that cater to the needs of visually challenged individuals. This makes it challenging for them to read and compose emails, leading to communication barriers and decreased independence. The need for an email client with voice-based input and output functionalities that cater to visually challenged individuals' needs is crucial.

Python Django, being a web framework, provides a suitable platform for developing voice-based email applications for visually challenged individuals. The framework has powerful features such as user authentication, user-friendly interfaces, and built-in security measures, which make it an ideal choice for developing a voice-based email application.

The primary objective of the project is to develop a voice-based email application using Python Django that caters to the needs of visually challenged individuals. The project aims to create an email client that is accessible and easy to use by visually challenged individuals, allowing them to manage their email accounts independently. The project will leverage voice recognition and text-to-speech technologies to enable visually challenged individuals to compose, send, and manage their emails using their voices.

The project's scope includes designing and developing a user-friendly interface that caters to the needs of visually challenged individuals. The interface will allow users to manage their email accounts, and compose and send emails using their voice. The project will leverage the SpeechRecognition library to convert the user's voice into text and the text-to-speech engine to convert emails into speech for the user to listen to. Additionally, the project will implement accessibility features such as high-contrast interfaces and error-handling mechanisms.

In conclusion, the voice-based email application for visually challenged individuals using Python Django aims to provide a solution that addresses the communication challenges faced by visually challenged individuals. The project's features such as voice input and output functionalities make it easier for visually challenged individuals to manage their email accounts independently. Python Django provides a suitable platform for developing a voice-based email application that is user-friendly, accessible, and secure. Ultimately, the project's objective is to improve the communication abilities of visually challenged individuals and enhance their independence.

1.1.1 Motivation

The motivation behind the voice-based email for visually challenged individuals project using Python Django is to address the communication challenges faced by visually challenged individuals in managing their email accounts. Traditional email clients and interfaces are not designed with accessibility features that cater to the needs of visually challenged individuals. This makes it challenging for them to read and compose emails, leading to communication barriers and decreased independence.

The voice-based email application aims to provide a solution that caters to visually challenged individuals' needs by leveraging voice recognition and text-to-speech technologies to enable them to compose, send, and manage their emails using their voice. The application provides an accessible and easy-to-use interface that caters to visually challenged individuals' needs, allowing them to manage their email accounts independently.

The motivation behind the project is to enhance visually challenged individuals' communication abilities and improve their quality of life by providing a platform that is accessible and user-friendly. The project aligns with the United Nations Convention on the Rights of Persons with Disabilities, which emphasizes the importance of accessibility and inclusivity for people with disabilities. By developing a voice-based email application that caters to visually challenged individuals' needs, the project promotes inclusivity, independence, and accessibility.

1.2 Objective

The first objective of the project is to provide visually challenged individuals with an accessible and user-friendly email client that meets their needs. This will involve designing an interface that is easy to use and navigate, with high-contrast colors and large fonts that are easy to read. The interface will be designed to provide users with clear and prominent labels that can be read out loud by text-to-speech technology. The interface will also be designed to provide feedback to users on their actions, making it easy for them to understand the status of their email accounts.

The project is to enable visually challenged individuals to compose, send, and manage their emails using their voices. This will involve integrating voice recognition technology into the email client that can accurately recognize the user's voice and convert it into text for email composition. The voice recognition system will be designed to handle various accents and speech patterns to ensure that it can be used by visually challenged individuals from different regions. Additionally, the system will be designed to allow users to review and edit the text of their emails before sending them.

Another objective of the project is to integrate text-to-speech technology that can convert emails into speech, enabling visually challenged individuals to listen to their emails. The text-to-speech technology will be designed to convert the text of the email into natural-sounding speech that is easy to understand. The text-to-speech system will also be designed to provide users with options to control the speed and volume of the speech output. This will allow users to listen to their emails at a comfortable pace and volume.

It ensures the security and privacy of users' email accounts. The system will be designed to store the users' email credentials securely and to use encryption to protect the privacy of their email communications. The system will also be designed to handle errors and provide users with feedback on the status of their email accounts to ensure that their emails are sent and received without errors.

1.3 Contribution

Voice-based email can be a very useful tool for visually challenged individuals, as it allows them to access and send emails without having to rely on visual interfaces. Additionally, the ability to translate emails into different languages, such as Malayalam, can further enhance accessibility for those who may be more comfortable communicating in their native language.
Here are some potential contributions of voice-based email with Malayalam translation for visually challenged individuals:

- Improved accessibility: By enabling individuals with visual impairments to access their emails through voice-based commands, this technology can significantly improve accessibility and empower individuals with disabilities to communicate more effectively.
- Increased independence: With the ability to access and send emails independently, individuals with visual impairments can become more self-sufficient and confident in their ability to communicate with others.
- Better communication: With the added feature of Malayalam translation, voice-based email can enable individuals who may not be fluent in English or who prefer to communicate in their native language to effectively communicate with others.
- Greater efficiency: Voice-based email with Malayalam translation can also streamline the process of composing and responding to emails, allowing individuals to save time and be more efficient in their communication.

Overall, the contribution of voice-based email with Malayalam translation for visually challenged individuals can be significant in terms of improving accessibility, increasing independence, enhancing communication, and boosting efficiency.

1.4 Report Organisation

The report on voice-based email for visually challenged individuals is organized into several sections. The report starts with an introduction that outlines the background and motivation for the project, as well as the objectives and contributions of the research. This section also includes a brief overview of the report organization, highlighting the various sections and their contents.

The report then delves into the literature survey, which provides an in-depth analysis of the existing research and technology in the field of voice-based email for visually challenged individuals. The methodology section follows, detailing the various steps taken to implement the project, including the workflow, and implementation process. The report also includes a section on Agile methodology, which outlines the user stories, product backlog, project plan, sprint backlog, and product backlog review. Finally, the report concludes with a discussion of the testing and validation process, as well as an overview of the Git version control system used throughout the project. Overall, the report is well-structured and comprehensive, providing a detailed account of the project from start to finish.

Chapter 2

Literature Survey

The proposed project aims to develop a voice-based email system that can help visually impaired people to use internet services for communication. The literature survey should focus on the existing research studies, systems, and technologies that have been developed to assist visually challenged individuals in using computers and accessing the internet. A comprehensive literature survey will help to identify the research gaps, opportunities, and challenges associated with the proposed project.

There are a few existing systems that offer voice-based email services for visually impaired people. One of the popular systems is the Thunderbird extension, which is an email client that has a built-in screen reader. The user can use this extension to read and write emails through voice commands. Another system is Microsoft Outlook Web Access (OWA), which has a built-in text-to-speech feature that can read out emails. However, these systems have limitations such as a lack of regional language support and limited functionalities.

The proposed system is a voice-based email system that focuses on providing basic functionalities like composing, reading, sending, and receiving emails along with voice-based assistants. The system is based on interactive voice response and eliminates the need for a keyboard or mouse. The speech is read in a regional language (Malayalam) to make it easy for blind people to understand. The system uses Python language to implement speech-to-text and text-to-speech. The proposed system aims to be user-friendly and efficient to use.

Chapter 3

Methodology

3.1 Introduction

The first step would be to conduct a needs assessment to determine the specific requirements and preferences of visually impaired individuals when it comes to email communication. This could involve conducting surveys or individual interviews with visually impaired individuals, as well as consulting with organizations that serve this population. This would be used to inform the design of the system, ensuring that it meets the needs of the intended users.

Once the design is finalized, the next step would be to develop the software for the voice-based email system. This would involve using programming languages like Python to implement speech-to-text and text-to-speech functionality, as well as developing an interactive voice response (IVR) system that allows users to navigate through the different functions of the email system using voice commands.

Finally, the system would need to be deployed and evaluated. This could involve conducting user testing with visually impaired individuals to gather feedback on the system's ease of use, functionality, and overall usefulness. Any necessary modifications or improvements could be made based on this feedback. It would also be important to evaluate the impact of the system on the daily lives of visually impaired individuals, and its potential to enhance their overall quality of life.

3.2 Workflow

The workflow for this project is given below.

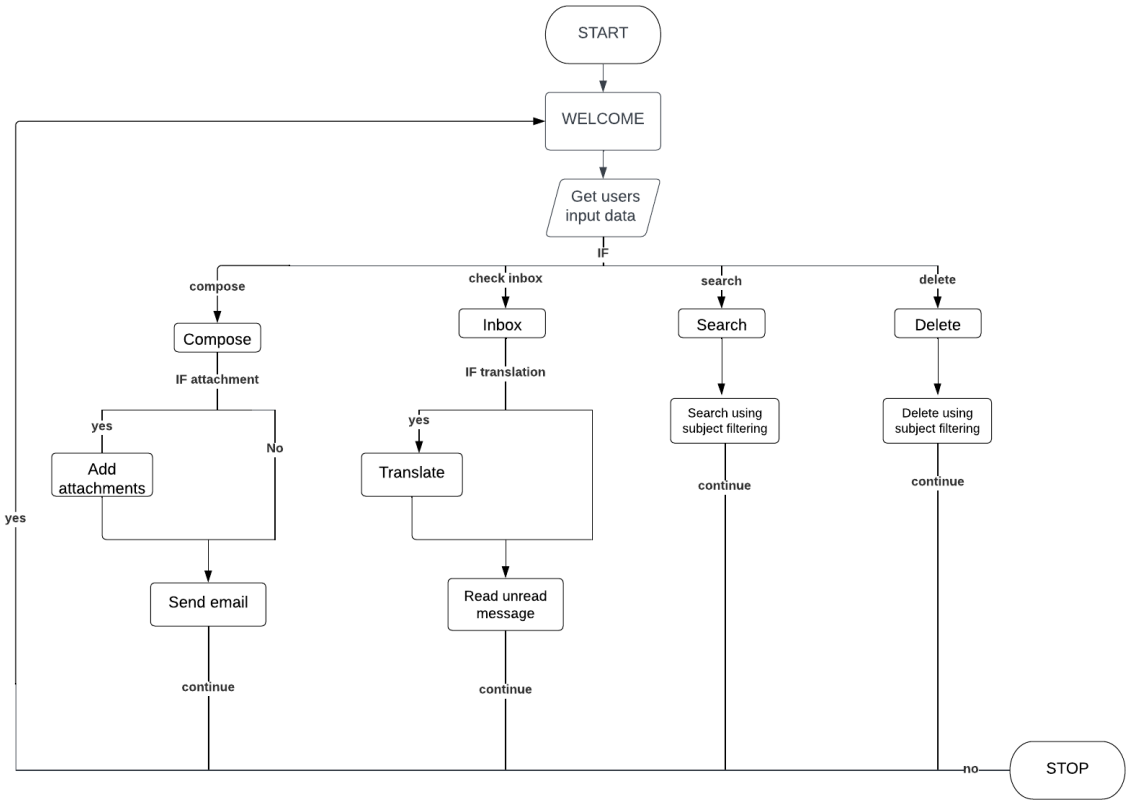


Figure 3.1: Workflow

3.3 Implementation

The implementation of the proposed voice-based email system for visually impaired individuals involves several steps. Firstly, the system requires a microphone for the user to input voice commands. These voice commands are then processed through a speech-to-text system, which uses the Python programming language to convert the spoken words into text. The text is then used to execute various email functions such as sending, receiving, and composing emails. Once the email function is performed, the system uses a text-to-speech system to read the email content back to the user in their regional language of Malayalam. This provides better understanding and accessibility for visually impaired individuals.

In addition to the basic email functionalities, the system also includes voice-based assistants to guide the user through the email process. This includes step-by-step instructions on how to perform various functions such as composing an email or sending a reply. The system also provides an option to save and store email addresses and contacts for future use. This eliminates the need for the user to remember the email addresses or type them in each time they wish to send an email.

The use of the Malayalam language for the text-to-speech function makes the system more accessible and user-friendly for visually impaired individuals in Kerala, India. This is especially important as Malayalam is the official language of Kerala and is widely spoken by its residents. The implementation of this feature not only makes the system more accessible but also helps promote inclusivity and equal access to technology for visually impaired individuals in the region.

Chapter 4

Agile Methodology

4.1 Introduction

Agile methodology is a popular approach to software development that is well-suited for the development of a voice-based email system for visually impaired people. Agile methodology is an iterative and incremental approach to software development that focuses on delivering working software quickly and continuously improving it over time. It is based on the principles of collaboration, flexibility, and rapid feedback. Agile methodology is particularly useful for software development projects that involve a high level of complexity and uncertainty, such as the development of the voice-based email system.

The agile methodology consists of several key practices that are designed to promote collaboration, rapid feedback, and continuous improvement. The first key practice is iterative development, which involves breaking the development process into small, manageable pieces or iterations. Each iteration involves a short period of time, typically one to four weeks, during which the development team works to complete a specific set of tasks or deliverables. At the end of each iteration, the team reviews the work completed and adjusts the development plan as necessary.

The second key practice is continuous integration, which involves integrating the work of different developers into a single, unified code base on a regular basis. This helps to ensure that the code is always up-to-date and that any issues or conflicts are identified and resolved quickly. The third key practice

is test-driven development, which involves writing automated tests for each new piece of code before it is integrated into the code base. This helps to ensure that the code is of high quality and that any defects are identified and corrected quickly.

In the case of the voice-based email system, the agile methodology can be particularly useful in ensuring that the system is developed quickly, efficiently, and in a manner that is responsive to the needs of visually impaired users. The iterative development approach can be used to break the development process into smaller, more manageable pieces, such as developing individual features or functions of the system. Continuous integration can be used to ensure that the code is always up-to-date and that any issues or conflicts are identified and resolved quickly. Test-driven development can be used to ensure that the system is of high quality and that any defects are identified and corrected quickly.

4.2 User Story

A user story is a concise, high-level description of a feature or functionality of a product that is written from the perspective of the user or customer. It is used to capture the user's needs, goals, and desired outcomes, rather than specifying the technical details of how the feature will be implemented. User stories are a key component of agile software development methodologies, which emphasize collaboration and responding to change over following a rigid plan.

User stories typically follow a simple format, such as "As a [user], I want [goal], so that [benefit]." The user story describes a specific feature or functionality that the user needs in order to achieve a particular goal or outcome. By focusing on the user's needs and goals, user stories help ensure that development efforts are aligned with the user's needs and that the resulting product is more likely to be useful and valuable.

User stories are often written on index cards or sticky notes and can be easily prioritized and organized by the development team. They are used to drive the development process, helping to ensure that the team is building features and functionality that are aligned with the user's needs. As the

product evolves and new requirements emerge, user stories can be added, updated, or removed, allowing the team to respond quickly and effectively to changing circumstances. Overall, user stories are an important tool for building better products that meet the needs and goals of the end users. The user story of the system is given in Table 4.1

User Story ID	**As a (type of user)**	**I want to**	**So that I can**
1	User	Access email using voice	Stay connected to people
2	User	Compose mail using voice	Sent emails without relying on text-based interfaces
3	User	Sent mail	Receive a confirmation message
4	User	Dictate the received mail	Use my voice to Obtain mail
5	User	Receive mail in regional language	Understand the mail content clearly
6	User	Reply to email	Respond to the received mail
7	User	Search	Obtain emails using voice
8	User	Detete	Remove emails using voice

Table 4.1: User Story

4.3 Product Backlog

The product backlog is a prioritized list of requirements or features that need to be developed for a product. It is a key artifact in agile software development methodologies, such as Scrum, and serves as the single source of truth for all the work that needs to be done on the product.

The product backlog is typically owned by the product owner, who is responsible for prioritizing the items on the backlog based on the needs and goals of the stakeholders. The items on the backlog are typically user stories,

which describe the features or functionality of the product from the perspective of the user.

The product backlog is dynamic and evolves over time as new information becomes available or as the product evolves. Items can be added, removed, or reprioritized based on feedback from stakeholders, changes in the market or technology, or other factors. The development team uses the product backlog to plan and execute their work, pulling items from the top of the backlog into each sprint or development cycle. The product backlog of the system is given in Table 4.3

ID	**Name**	**Priority (High/Medium /Low)**	**Estimate (Hr)**	**Status (Planned/ Progressing/ Completed)**
1	Voice recognition	High	75	Completed
2	Email configuration	High	30	Completed
3	Email composition	High	40	Completed
4	Email delete	High	40	Completed
5	Email reading	High	65	Completed
6	Reply email	Medium	24	Completed
7	Language translation	High	50	Completed

Table 4.2: Product Backlog

4.4 Project Plan

A project plan is a detailed document that outlines the objectives, tasks, timelines, resources, and potential risks associated with a specific project. It serves as a roadmap that guides the project team in executing and completing the project successfully within the defined constraints, such as budget, timeline, and scope.

A project plan is critical for ensuring that a project is completed efficiently, effectively, and on time. It helps project managers and team members stay focused on the project goals and objectives and provides a framework for managing tasks, timelines, and resources. A well-defined project plan also helps identify potential risks and challenges, allowing project teams to prepare and implement appropriate strategies to mitigate them. Ultimately, a project plan is essential for managing a project and ensuring that it meets or exceeds stakeholders' expectations.The product plan of the system is given in Table 4.4

User Story ID	**Task Name**	**Start Date**	**End Date**	**Days**	**Status**
1	Sprint 1	01/02/2023	10/02/2023	10	Completed
2		11/02/2023	21/02/2023	11	Completed
3	Sprint 2	27/07/2023	08/03/2023	10	Completed
4		09/03/2023	20/03/2023	11	Completed
5	Sprint 3	22/03/2023	31/03/2023	10	Completed
6		01/04/2023	11/04/2023	11	Completed
7	Sprint 4	15/04/2023	24/04/2023	10	Completed
8		25/04/2023	05/05/2023	11	Completed

Table 4.3: Project Plan

4.5 Sprint Backlog (Plan)

The Sprint Backlog is a plan created by the Development Team at the beginning of each Sprint. It details the work that needs to be completed during the Sprint to meet the Sprint Goal. The Sprint Backlog is created during the Sprint Planning event, where the Development Team decides on the work that will be completed during the Sprint. The Product Owner presents the Product Backlog items that they want to be completed during the Sprint, and the Development Team determines how much of the work they can re-

alistically complete during the Sprint. The detailed sprint backlog (Plan) is given below.

Backlog item	Completion date	Original estimate	Day 1 01/02/2023	Day 2 02/02/2023	Day 3 03/02/2023	Day4 04/02/2023	Day 5 05/02/2023	Day 6 06/02/2023	Day 7 07/02/2023	Day 8 08/02/2023	Day 9 09/02/2023	Day 10 10/02/2023
User Story #1		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	10/02/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	05/02/2023	5	1	1	1	1	1	0	0	0	0	0
Testing	07/02/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 11/02/2023	Day 12 12/02/2023	Day 13 13/02/2023	Day 14 14/02/2023	Day 15 15/02/2023	Day 16 16/02/2023	Day 17 17/02/2023	Day 18 18/02/2023	Day 19 19/02/2023	Day 20/2023 20/02/2023	Day 21 21/02/2023
User Story #2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	21/02/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	12/02/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	18/0220/23	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	1	2	1	2	2	4	1	3	2	1

Table 4.4: Sprint Backlog (Plan) - Sprint 1

Backlog item	Completion date	Original estimate	Day 1 27/02/2023	Day 2 28/02/2023	Day 3 01/03/2023	Day4 02/02/2023	Day 5 03/03/2023	Day 6 04/03/2023	Day 7 05/03/2023	Day 8 06/03/2023	Day 9 07/03/2023	Day 10 08/03/2023
User Story #3		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	08/03/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	03/03/2023	5	1	1	1	1	1	0	0	0	0	0
Testing	05/03/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 09/03/2023	Day 12 10/03/2023	Day 13 11/03/2023	Day 14 12/03/2023	Day 15 13/03/2023	Day 16 14/03/2023	Day 17 15/03/2023	Day 18 16/03/2023	Day 19 17/03/2023	Day 20 18/03/2023	Day 21 19/03/2023
User Story #4		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	19/03/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	13/03/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	16/03/2023	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	2	2	2	1	3	2	2	2	2	1

Table 4.5: Sprint Backlog (Plan) - Sprint 2

Backlog item	Completion date	Original estimate	Day 1 22/03/2023	Day 2 23/03/2023	Day 3 24/03/2023	Day4 25/02/2023	Day 5 26/03/2023	Day 6 27/03/2023	Day 7 28/03/2023	Day 8 29/03/2023	Day 9 30/03/2023	Day 10 31/03/2023
User Story #5		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	31/03/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	26/03/2023	5	1	1	1	1	1	0	0	0	0	0
Testing	28/03/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 01/04/2023	Day 12 02/04/2023	Day 13 03/04/2023	Day 14 04/04/2023	Day 15 05/04/2023	Day 16 06/04/2023	Day 17 07/04/2023	Day 18 08/04/2023	Day 19 09/04/2023	Day 20 10/04/2023	Day 21 11/04/2023
User Story #6		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	11/04/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	06/04/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	08/04/2023	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	2	1	2	2	3	2	2	2	2	1

Table 4.6: Sprint Backlog (Plan) - Sprint 3

Backlog item	Completion date	Original estimate	Day 1 15/04/2023	Day 2 16/04/2023	Day 3 17/04/2023	Day4 18/04/2023	Day 5 19/04/2023	Day 6 20/04/2023	Day 7 21/04/2023	Day 8 22/04/2023	Day 9 23/04/2023	Day 10 24/04/2023
User Story #7		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	24/04/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	19/04/2023	5	1	1	1	1	1	0	0	0	0	0
Testing	21/04/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 25/04/2023	Day 12 26/04/2023	Day 13 27/04/2023	Day 14 28/04/2023	Day 15 29/04/2023	Day 16 30/04/2023	Day 17 01/05/2023	Day 18 02/05/2023	Day 19 03/05/2023	Day 20 04/05/2023	Day 21 05/05/2023
User Story #8		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	05/05/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	30/04/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	02/05/2023	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	2	2	2	2	2	2	2	2	2	2

Table 4.7: Sprint Backlog (Plan) - Sprint 4

4.6 Sprint Backlog (Actual)

The Sprint Backlog is a subset of the Product Backlog, consisting of items that the Development Team has committed to completing during a Sprint. It is created during the Sprint Planning meeting, where the Development Team selects the highest priority Product Backlog items that they can complete within the upcoming Sprint. The Sprint Backlog is a plan for how the Development Team will accomplish the selected Product Backlog items and achieve the Sprint Goal. The detailed sprint backlog (Actual) is given below.

Backlog item	Completion date	Original estimate	Day 1 01/02/2023	Day 2 02/02/2023	Day 3 03/02/2023	Day4 04/02/2023	Day 5 05/02/2023	Day 6 06/02/2023	Day 7 07/02/2023	Day 8 08/02/2023	Day 9 09/02/2023	Day 10 10/02/2023
User Story #1		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	10/02/23	3	0	0	0	0	0	0	0	1	1	1
Coding	08/02/23	5	1	0	1	1	1	0	1	0	0	0
Testing	07/02/23	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 11/02/2023	Day 12 12/02/2023	Day 13 13/02/2023	Day 14 14/02/2023	Day 15 15/02/2023	Day 16 16/02/2023	Day 17 17/02/2023	Day 18 18/02/2023	Day 19 19/02/2023	Day 20/2023 20/02/2023	Day 21 21/02/2023
User Story #2		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	21/02/23	3	0	0	0	0	0	0	0	0	1	1	1
Coding	13/02/23	6	1	1	1	0	1	1	1	0	0	0	0
Testing	19/02/23	2	0	0	0	0	0	0	1	0	1	0	0
Total		21	2	1	2	1	2	2	4	1	3	2	1

Table 4.8: Sprint Backlog (Actual) - Sprint 1

Backlog item	Completion date	Original estimate	Day 1 27/02/2023	Day 2 28/02/2023	Day 3 01/03/2023	Day4 02/02/2023	Day 5 03/03/2023	Day 6 04/03/2023	Day 7 05/03/2023	Day 8 06/03/2023	Day 9 07/03/2023	Day 10 08/03/2023
User Story #3		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	08/03/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	04/03/2023	5	1	1	1	1	0	1	0	0	0	0
Testing	05/03/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 09/03/2023	Day 12 10/03/2023	Day 13 11/03/2023	Day 14 12/03/2023	Day 15 13/03/2023	Day 16 14/03/2023	Day 17 15/03/2023	Day 18 16/03/2023	Day 19 17/03/2023	Day 20 18/03/2023	Day 21 19/03/2023
User Story #4		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	19/03/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	13/03/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	16/03/2023	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	2	2	2	1	3	2	2	2	2	1

Table 4.9: Sprint Backlog (Actual) - Sprint 2

Backlog item	Completion date	Original estimate	Day 1 22/03/2023	Day 2 23/03/2023	Day 3 24/03/2023	Day4 25/02/2023	Day 5 26/03/2023	Day 6 27/03/2023	Day 7 28/03/2023	Day 8 29/03/2023	Day 9 30/03/2023	Day 10 31/03/2023
User Story #5		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	31/03/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	28/03/2023	5	1	1	0	1	1	0	1	0	0	0
Testing	28/03/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 01/04/2023	Day 12 02/04/2023	Day 13 03/04/2023	Day 14 04/04/2023	Day 15 05/04/2023	Day 16 06/04/2023	Day 17 07/04/2023	Day 18 08/04/2023	Day 19 09/04/2023	Day 20 10/04/2023	Day 21 11/04/2023
User Story #6		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	11/04/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	06/04/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	08/04/2023	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	2	1	2	2	2	3	2	2	2	1

Table 4.10: Sprint Backlog (Actual) - Sprint 3

Backlog item	Completion date	Original estimate	Day 1 15/04/2023	Day 2 16/04/2023	Day 3 17/04/2023	Day4 18/04/2023	Day 5 19/04/2023	Day 6 20/04/2023	Day 7 21/04/2023	Day 8 22/04/2023	Day 9 23/04/2023	Day 10 24/04/2023
User Story #7		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	24/04/2023	3	0	0	0	0	0	0	0	1	1	1
Coding	19/04/2023	5	1	1	1	1	1	0	0	0	0	0
Testing	21/04/2023	2	0	0	0	0	0	1	1	0	0	0

Backlog item	Completion date	Original estimate	Day 11 25/04/2023	Day 12 26/04/2023	Day 13 27/04/2023	Day 14 28/04/2023	Day 15 29/04/2023	Day 16 30/04/2023	Day 17 01/05/2023	Day 18 02/05/2023	Day 19 03/05/2023	Day 20 04/05/2023	Day 21 05/05/2023
User Story #8		Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Form Design	05/05/2023	3	0	0	0	0	0	0	0	0	1	1	1
Coding	30/04/2023	6	1	1	1	1	1	1	0	0	0	0	0
Testing	02/05/2023	2	0	0	0	0	0	0	1	1	0	0	0
Total		21	2	2	2	2	2	2	2	2	2	2	1

Table 4.11: Sprint Backlog (Actual) - Sprint 4

4.7 Product Backlog Review

Product Backlog Review is an essential ceremony in Agile software development, where the Product Owner and the Development Team collaborate to review the product backlog items. The primary goal of the Product Backlog Review is to ensure that the product backlog is up-to-date, prioritize the items, and provide a shared understanding of the product backlog among the team. The Product backlog review is given below.

REVIEW FORM

SPRINT 1

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
1	Voice recognition accuracy could be improved	Easy access through voice
2	Confirm message can be added	Include attachments while sending emails if necessary

Table 4.12: Product Backlog Review - Sprint 1

REVIEW FORM

SPRINT 2

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
3	Satisfying result	Satisfying result
4	Dictate the email in Malayalam	Tell the number of unread messages

Table 4.13: Product Backlog Review - Sprint 2

REVIEW FORM

SPRINT 3

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
5	Improve accuracy when converting to regional language	Improve accuracy when converting to regional language
6	Satisfying result	Satisfying result

Table 4.14: Product Backlog Review - Sprint 3

REVIEW FORM

SPRINT 4

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
7	Searching can be done using a subject filtering	Easy search
8	Deleting can be done using a subject filtering	Easy Delete

Table 4.15: Product Backlog Review - Sprint 4

4.8 Sprint Review

Sprint Review is a meeting that takes place at the end of each sprint in an Agile Scrum project. The purpose of the Sprint Review is to inspect the Increment produced during the sprint, gather feedback, and adapt the Product Backlog if needed. The Sprint Review is attended by the Scrum Team, stakeholders, and customers, who review the Increment and provide feedback. Detailed sprint review is given below.

REVIEW FORM

SPRINT 1

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
1	Accuracy improved successfully	Satisfying result
2	Confirm message successfully added	Attachments can be added successfully

Table 4.16: Sprint Review - Sprint 1

REVIEW FORM

SPRINT 2

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
3	Satisfying result	Satisfying result
4	Emails dictated in Malayalam is added successfully	The number of unread messages are added successfully

Table 4.17: Sprint Review - Sprint 2

REVIEW FORM

SPRINT 3

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
5	Partially Improved accuracy when converting to regional language	Satisfying result
6	Satisfying result	Satisfying result

Table 4.18: Sprint Review - Sprint 3

REVIEW FORM

SPRINT 4

User Story ID	**Comments from Scrum master if any**	**Comments from Product Owner if any**
7	Searching done using a subject filtering successfully	Satisfying result
8	Deleting done using a subject filtering successfully	Satisfying result

Table 4.19: Sprint Review - Sprint 4

4.9 Testing and Validation

Testing and validation are essential parts of any software development process. In the case of the voice-based email system for visually challenged people, testing and validation are crucial to ensure that the system is working as intended and meeting the requirements of the end-users.The testing process should begin with unit testing, where each module of the system is tested independently to ensure that it performs its functions correctly. The next step is integration testing, where the different modules are combined and tested as a single system. This testing process helps to identify any issues that may arise when the modules interact with each other.

Once the integration testing is complete, the system should undergo system testing to ensure that it meets all the functional and non-functional requirements. This testing process should also include testing the system's accessibility features to ensure that visually challenged people can use the system without difficulty.

Test#	Date	Action	Expected Result	Actual Result	Pass ? <Yes/No>
1	10/02/2023	Access email using voice	Email account has to be accessed	Email account is accessed using proper configuration	Yes
2	18/02/2023	Compose mail using voice	Email has to be sent	Email has been sent successfully	Yes

Table 4.20: Test Sprint 1

Test#	Date	Action	Expected Result	Actual Result	Pass ? <Yes/No>
3	08/03/2023	Sent mail	To receive a confirmation message	Confirmation message received successfully	Yes
4	19/03/2023	Compose mail using voice	Dictate received mail	Properly dictated	Yes

Table 4.21: Test Sprint 2

Test#	Date	Action	Expected Result	Actual Result	Pass ? <Yes/No>
5	31/03/2023	Receive mail in regional language	Mail read in Malayalam	Mail is translated to malayalam successfully	Yes
6	11/04/2023	Reply to email	To sent the response	Response can be sent	Yes

Table 4.22: Test Sprint 3

Test#	Date	Action	Expected Result	Actual Result	Pass ? <Yes/No>
7	24/04/2023	Search	Search mail	Search mail using subject	Yes
8	05/05/2023	Delete	Delete mail	Delete mail using subject	Yes

Table 4.23: Test Sprint 4

4.10 Git

Git is a version control system used to track and manage changes in software code. In the context of the voice-based email system for visually impaired individuals, Git can be used to manage the project's source code and track changes made during the development process.

When working on the project, developers can use Git to create a local repository on their machine to store the project's code. They can make changes to the code, and then use Git to stage and commit those changes. Each commit represents a snapshot of the code at a specific point in time.

Once the developers have made several changes and commits, they can push their changes to a remote repository hosted on a platform like GitHub. This allows other team members to access the latest code changes and collaborate on the project. With Git, it's easy to track the history of the project, identify changes made by different team members, and revert to previous versions if necessary.

Chapter 5

Conclusion

In conclusion, the proposed project aims to develop a voice-based email system for visually impaired individuals to enhance their communication abilities. The project has been developed using the Python programming language, which incorporates text-to-speech and speech-to-text functionalities to create an interactive voice response system that eliminates the need for a keyboard or mouse input. This will allow visually challenged users to give voice commands to compose, read, send, and receive emails efficiently.

The project has been implemented following the agile methodology, which has allowed for continuous iterations and improvements to be made. The testing and validation process involved multiple rounds of user testing, where visually impaired individuals provided valuable feedback that was used to refine the system further. The system was also tested for its accuracy, efficiency, and reliability in handling voice commands.

The use of a regional language (Malayalam) in the system has made it easily understandable for the target audience. The integration of Git has enabled version control and has facilitated the development process of the project. The successful implementation of the project will provide visually impaired individuals with a tool that will enable them to communicate effectively over the internet, promoting the inclusion and empowerment of visually challenged individuals in a growing digital India.

Bibliography

[1] B. V. Patil and K. Sreelakshmi, *"Implementation of Voice Based E-Mail System for Visually Challenged," 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 2022, pp. 1-9, doi: 10.1109/INCOFT55651.2022.10094386..*

[2] P. A. Tiwari, P. Zodawan, H. P. Nimkar, T. Rotke, P. G. Wanjari and U. Samarth, *"A Review on Voice based E-Mail System for Blind," 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020, pp. 435-438, doi: 10.1109/ICICT48043.2020.9112539.*

Appendix A

Source Code

```
{% load static %}
<!DOCTYPE html>
<html>
  <head>
    <title>Mail ID Form</title>
    <style>
      body {
        margin: 0;
        padding: 0;
        /* background-color: rgba(0,0,0,0.1); */
        background-image: url;
        background-size: cover;
        background-position: center;
        background-attachment: fixed;
      }

      .form-container {
        max-width: 500px;
        margin: 50px auto;
        padding: 20px;
        background-color: rgba(255,255,255,0.1);
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0,0,0,0.2);
      }
```

```
        .composeMail {
          text-align: center;
          color: #555;
          margin-bottom: 30px;
          text-shadow: 1px 1px 1px white, -1px -1px 1px
              white, 1px -1px 1px white, -1px 1px 1px
              white;
        }

        label {
          display: block;
          margin-bottom: 10px;
          color: #0d0d0d;
          font-size: 20px;
          text-shadow: 2px 2px 4px rgba(255, 255, 255, 1)
              ;
        }

        {
          width: 97%;
          padding: 10px;
          border: 2px solid rgba(0, 0, 0, 0.2);
          border-radius: 4px;
          box-shadow: inset 0 0 5px rgba(0,0,0,0.2);
          margin-bottom: 20px;
          font-size: 16px;
          color: #555;
          background: transparent;
      }
.hidden-button {
          display: none;
    }

   </style>
  </head>
  <body>
    <div class=navbar>
        <h2>Voice Based Email for Visually Challenged <
```

```
            /h2>


    <div class=form-container>
      <h1 class=composeMail>MESSAGE DELETED
          SUCCESSFULLY</h1>


    </div>
  </body>

</html>
```

Appendix B

Output

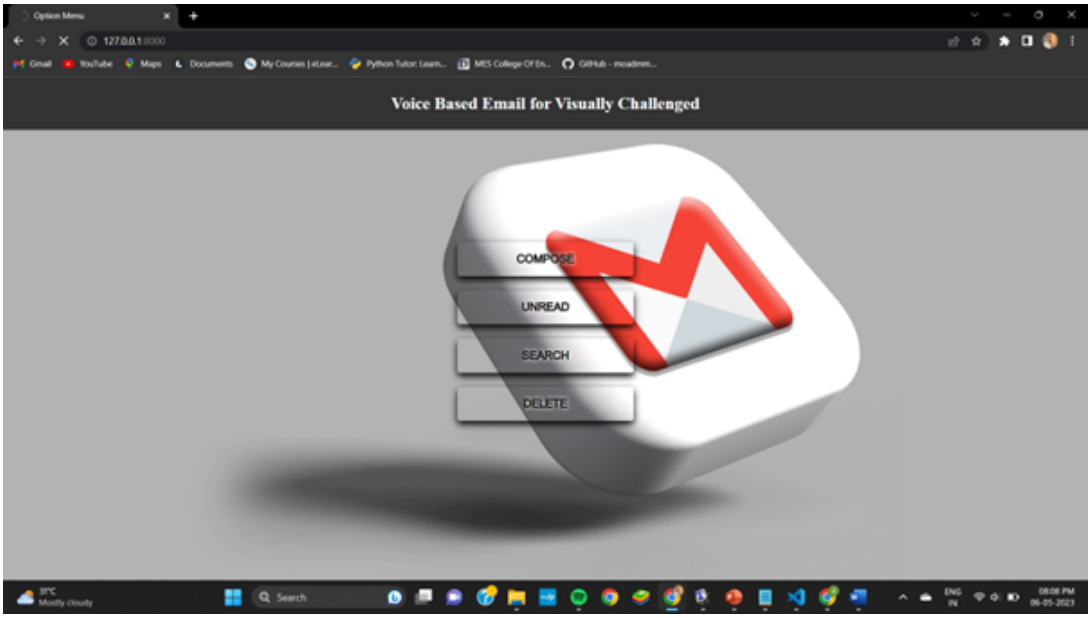


Figure B.1: User Interface 1

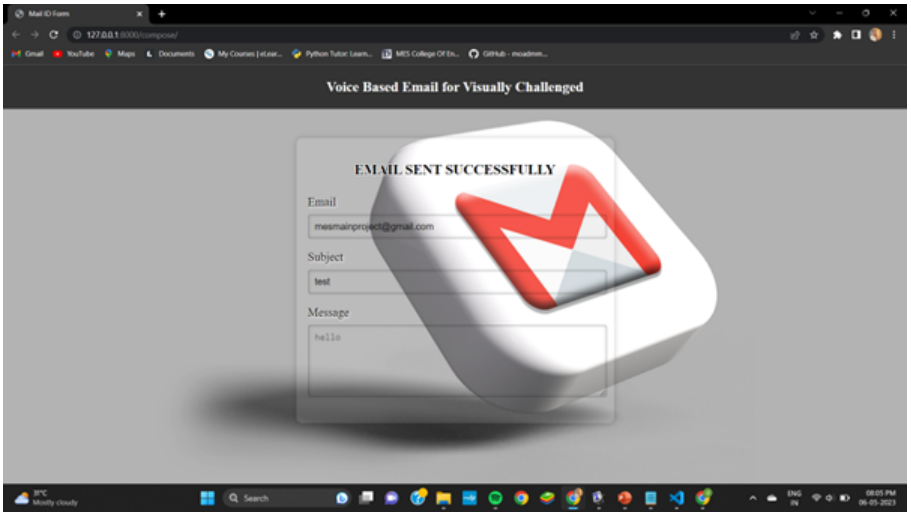


Figure B.2: User Interface 2

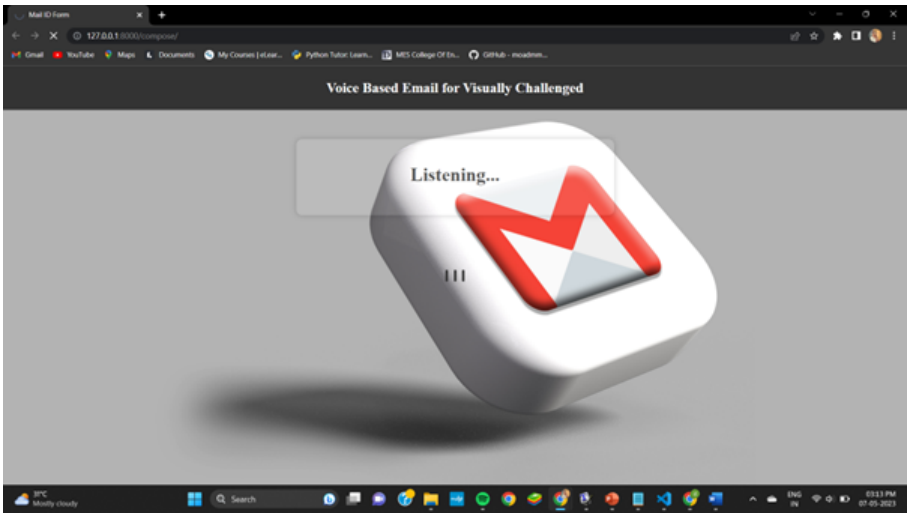


Figure B.3: User Interface 3

Appendix C

Git History

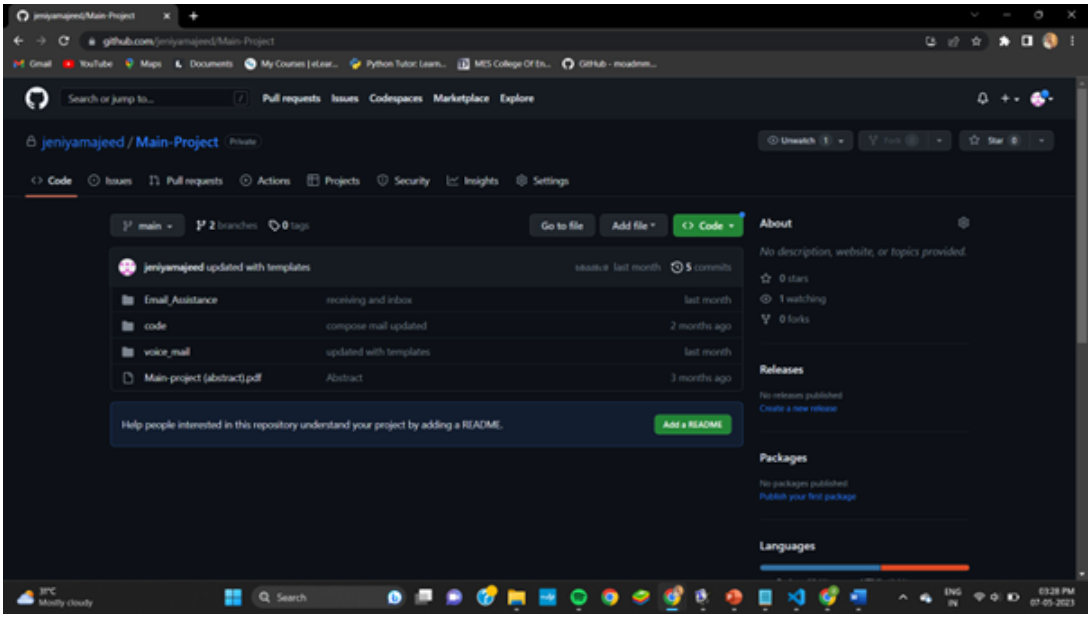


Figure C.1: Git Commits