VAW – Voice Assistant for Windows

A Project Report
Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Jaikumar Gaja
TIT2223022

Under the esteemed guidance of
Ms. Biju Ramesh

DEPARTMENT OF INFORMATION TECHNOLOGY
SIES COLLEGE OF ARTS, SCIENCE & COMMERCE (AUTONOMOUS)
SION (W), MUMBAI, 400022
MAHARASHTRA

2022

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

Roll No**:** TIT2223022

1.	Name of the Student	Jaikumar Gaja
2.	Title of the Project for Windows	VAW – Voice Assistant
3.	Name of the Guide	Ms. Biju Ramesh
4.	Teaching experience of the Guide	19 years
5.	Is this your first submission?	Yes ☑ No ☐

Jaikumar Gaja

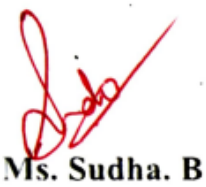
Signature of the Student

Date: 20/10/2022

Ms. Biju Ramesh

Signature of the Guide

Date: 20/10/2022



Ms. Sudha. B

Signature of the Coordinator

Date: 20/10/2022

SIES COLLEGE OF ARTS, SCIENCE & COMMERCE
(AUTONOMOUS)
(Affiliated to University of Mumbai)
SION (W), MUMBAI-400022

DEPARTMENT OF INFORMATION TECHNOLOGY

SIES College of Arts, Science & Commerce
RISE WITH EDUCATION
Sion (West), Mumbai – 400022.
(Autonomous)

CERTIFICATE

This is to certify that the project entitled, **"VAW – Voice Assistant for Windows"**, is bonafide work of **Mr. Jaikumar Gaja** bearing Seat No: **TIT2223022 s**ubmitted in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY.**

Ms. Biju Ramesh
Internal Guide

Ms. Sudha B
Coordinator

Date: 20/10/2022

20 OCT 2022

External Examiner

ABSTRACT

At this present time, day-to-day life is becoming smarter and more integrated with technology. There are some voice assistants like Google, Siri, Cortana, and Alexa that we are already familiar with. Now in our voice assistant system, it can act as a daily schedule reminder, note writer, calculator, search tool, and perform system operating tasks like closing an open application and shutting down or restarting the PC. This project works on voice input and gives voice output and displays the text on the screen. It is the primary goal of our voice assistance to make people more efficient and to provide them with instant computed results. The voice assistant takes the voice input through our microphone and converts our voice into computer-understandable language and gives the required solutions and answers the questions that were asked by the user. This voice assistant connects with the World Wide Web to provide results that the user has questioned.

Python is used in this project because it is compatible and it also has several library files that go well with a voice assistant. APIs are also used to access different services like YouTube, Spotify, and Twitter right from this application.

ACKNOWLEDGEMENT

I want to thank Mrs. Biju Ramesh for her guidance while creating the project. She was always available for help and advice. Her guidance helped me to allocate time for the project. When I had any problems with the project, she helped me solve them, which made it easier for me to finish the project. I am thankful to her for all her support.

I also want to thank all my friends, teachers, and sister for giving me ideas and helping me find solutions to problems. They all were very supportive and helpful. I am thankful to them.

DECLARATION

I hereby declare that the project entitled, "**VAW – Voice Assistant for Windows**" done at **SIES COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS)**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Jaikumar Gaja

Name and Signature of the Student

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1: Background:

"Cortana" is an already existing voice assistant embedded in Windows. It is personal voice assistant software that interacts with the user through voice, recognizes commands, and acts on them. It learns to adapt to the user's speech and thus, improves voice recognition over time. It integrates with the calendar, contacts, and applications on the device.

Cortana has some limitations, including the inability to close any open application or to shut down or restart the PC. So, in VAW these drawbacks of Cortana will be eliminated.

1.2: Objectives:

Building VAW aims to resolve Cortana's drawbacks, such as the need to restart the PC or close applications, and to make it easier to use for the users. With this project, an environment will be created where a user can do the tasks just by saying the task to the computer. The primary objective of this project is to use the services of different applications and websites through API. With VAW, a user will be able to search the web, extract weather data and help with vocabulary.

1.3: Purpose, Scope, and Applicability:

1.3.1: Purpose:

The purpose of VAW is to interact with the user and take actions according to the user's request like making to-do lists, setting alarms, providing weather reports, sports scores, and other real-time information, such as news and stock-market information in this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized. It can be seen that VAW can be a good example for moving towards faster operations.

1.3.2: Scope:

This project will be handy for basic tasks like setting alarms, creating a to-do list, creating a reminder, sending e-mails, checking the weather report, searching for news, and shutting down or restarting the PC. Users can also manage time with this application by doing basic tasks through this application. The user just has to say the correct phrase and the application will do the work

As a limitation, voice recognition may not be accurate since some people may have a different accent from others. So, a user should be specific with their commands so that the system recognizes them and takes the appropriate action.

1.3.3: Applicability

It can also be used by individuals with no prior IT experience. They just have to say the commands and the Application will fetch them the results or give a confirmation about the task. Sometimes a user may be too lazy to click buttons and type keywords. Therefore, rather than manually entering the keywords they want to search for, the user can use the application's search feature. Following that, the user will receive the search results related to their command.

Chapter 2: Survey of Technologies

2.1 Front-End and Back-End Development

2.1.1 Java

Java is an object-oriented language, which means all programs are made of entities representing concepts or physical things known as “objects”. Java programs are found in desktops, servers, mobile devices, smart cards and Blu-ray Discs (BD).

Many desktop applications are developed in Java. Swing, Abstract Windowing Toolkit (AWT) and JavaFX are the main tools used for easy GUI development.

Pros: -

- Simplicity.
- Cross platform
- Provides a myriad of APIs to accomplish various necessary tasks such as networking, XML parsing, utilities, and database connection
- Supports distributed computing and multithreading
- Extremely secure because of the implementation of a security manager that is used to define class access
- Open-source library
- Rich ecosystem
- Robust and scalable

Cons: -

- Lack of the availability of templates limits the development of data structures of high quality.
- Memory management is expensive.
- Java is slow as compared to other languages like C and C++
- No backup facility
- Verbose and complex code
- Poor GUI

2.1.2 Kotlin

Kotlin is a general purpose, free, open source, statically typed "pragmatic" programming language initially designed for the JVM (Java Virtual Machine) and Android that combines object-oriented and functional programming features. It is focused on interoperability, safety, clarity, and tooling support. Versions of Kotlin targeting JavaScript ES5.1 and native code (using LLVM) for a number of processors are in production as well.

Pros

- Most importantly, these are some advantages that Kotlin offers.
- Being easy to use, Kotlin boosts productivity. The majority of the available IDEs in the present market support Kotlin code. This also helps improve efficiency as developers don't need to learn new IDEs for Android projects.
- Kotlin offers great interoperability with Java. This allows developers to switch to Java from Kotlin or vice versa extremely easier. Kotlin offers great consistency with Java and all Java-based frameworks and tools used by Android developers.
- As tried and tested language, Kotlin offers consistency. It provides unmatched reliability among languages that have been launched in the recent years.
- Kotlin offers a very intuitive and clean syntax. It ultimately results in fewer coding errors and more efficient programming output on the part of developers. Mobile developers can actually complete more tasks in less time by using Kotlin.

Cons

- On the other hand, these are few drawbacks of Kotlin.
- Code compilation in Kotlin shows fluctuation. This is especially evident in the incremental development tasks.
- As it is a new language, so Kotlin doesn't have large talent pool of developers.
- Support of a big developer community is also what Kotlin lacks.
- Kotlin in comparison to Java, offers few learning resources and tutorials.

2.1.3 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Pros

- Presence of third-party modules
- Extensive support libraries (NumPy for numerical calculations, Pandas for data analytics, etc.)
- Dynamically typed language (No need to mention data type based on the value assigned, it takes data type)
- Ideal for prototypes – provide more functionality with less coding
- Python’s clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities, as well as its own unit testing framework, which makes it more efficient.
- Portable across Operating systems

Cons

- Python is an interpreted language, and it is slow compared to C/C++.
- For any memory intensive tasks Python is not a good choice.
- Python is not memory efficient
- Weak in mobile-computing
- The Python’s database access layer is primitive and underdeveloped in comparison to the popular technologies like JDBC and ODBC.

2.1.4 Flutter

Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, desktop and embedded devices from a single codebase. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia and the web from a single codebase. Flutter works with existing code, is used by developers and organizations around the world and is free and open source.

Pros:

- the app UI and logic don't change depending on the platform.
- faster code development.
- increased time-to-market speed
- close to native app performance
- enormous UI customization potential
- separate rendering engine • no reliance on platform-specific UI components
- suitable for any target platform

Cons

- The one disadvantage is that Flutter apps are bigger in size. Generally, Flutter apps are greater than 4MB.
- Apps built by Flutter are quite enjoyable in Android but face some difficulties in iOS. Google is working on it as they are looking forward to capturing the USA's iPhone/iOS app development market.
- Flutter doesn't support third-party libraries and sometimes increases the programming task. However, if you plan an app properly, then Flutter can become the best.
- Flutter does not support Android TV and Apple TV.
- Though Dart Language is suitable for Flutter, its challenging nature sometimes makes coding difficult. But you can overcome this with the support of Java, C++, C#, and Objective-C.

2.2 Database

2.2.1 PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system. It is free and open-source relational database management system, maintained by PostgreSQL Global Development Group and its prolific community. PostgreSQL seems to be more universal. It is widely available on multiple operating systems: FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows. PostgreSQL has user-defined functions in proprietary language PL/pgSQL or with common languages like Perl, Python, Tcl etc. PostgreSQL can be driven entirely from the command line. PostgreSQL has a better concurrency management system. It handles very well the case where multiple processes can access and modify shared data at the same time.

Pros

- PostgreSQL can run dynamic websites and web apps as a LAMP stack option.
- PostgreSQL's write-ahead logging makes it a highly fault-tolerant database.
- PostgreSQL source code is freely available under an open-source license. This allows you the freedom to use, modify, and implement it as per your business needs.
- PostgreSQL supports geographic objects so you can use it for location-based services and geographic information systems.
- PostgreSQL supports geographic objects so it can be used as a geospatial data store for location-based services and geographic information systems.
- Low maintenance and administration for both embedded and enterprise use of PostgreSQL.

Cons

- Below are the disadvantages/limitations of PostgreSQL:
- Postgres is not owned by one organization. So, it has had trouble getting its name out there despite being fully featured and comparable to other DBMS systems
- Changes made for speed improvement requires more work than MySQL as PostgreSQL focuses on compatibility
- Many open-source apps support MySQL, but may not support PostgreSQL
- On performance metrics, it is slower than MySQL.

2.2.2 MongoDB

MongoDB is a non-relational document database that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables you to store unstructured data, and it provides full indexing support, and replication with rich and intuitive APIs.

Pros

- Flexible Database: We know that MongoDB is a schema-less database. That means we can have any type of data in a separate document. This thing gives us flexibility and a freedom to store data of different types.
- High Speed: MongoDB is a document-oriented database. It is easy to access documents by indexing. Hence, it provides fast query response. The speed of MongoDB is 100 times faster than the relational database.
- High Availability: MongoDB has features like replication and gridFS. These features help to increase data availability in MongoDB. Hence the performance is very high.
- Scalability: A great advantage of MongoDB is that it is a horizontally scalable database. When you have to handle a large data, you can distribute it to several machines.
- Ad-hoc Query Support: MongoDB has a very advanced feature for ad hoc queries. This is why we don't need to worry about fore coming queries coming in the future.
- Easy Environment Setup: It is easier to setup MongoDB then RDBMS. It also provides JavaScript client for queries.

Cons

- Joins not Supported: MongoDB doesn't support joins like a relational database. Yet one can use joins functionality by adding by coding it manually. But it may slow execution and affect performance.
- High Memory Usage: MongoDB stores key names for each value pairs. Also, due to no functionality of joins, there is data redundancy. This results in increasing unnecessary usage of memory.
- Limited Data Size: You can have document size, not more than 16MB.
- Limited Nesting: You cannot perform nesting of documents for more than 100 levels.

2.2.3 MySQL

MySQL was created by a Swedish company MySQL AB. The features are like support to cross-platform, stored procedures, triggers, cursors, data definition language, ACID compliance, SSL support, views updatable, partitioning, Indexing, select, commit grouping, Unicode support and many more. There are certain limitations in My SQL. In MySQL, Triggers are limited to only one action per timing. It means only one trigger can be executed on the table if any event happens on the table. Triggers cannot be defined on views as well. The other limitation is MySQL does not follow the full SQL standards. MySQL uses the 'mysqldump' backup tool, which supports backing up of data from all the storage engines. The other MySQL back up software program is 'XtraBackup', which is open-source. MySQL can be run on Cloud as well as Amazon and Microsoft Azure. MySQL can be used as a service.

Pros

- MySQL is more secure as it consists of a solid data security layer to protect sensitive data from intruders and passwords in MySQL are encrypted.
- MySQL is available for free to download and use from the official site of MySQL.
- MySQL is compatible with most of the operating systems, including Windows, Linux, NetWare, Novell, Solaris and other variations of UNIX.
- MySQL provides the facility to run the clients and the server on the same computer or on different computers, via internet or local network.
- MySQL has a unique storage engine architecture which makes it faster, cheaper and more reliable.
- MySQL has a client-server architecture. There can be any number of clients or application programs which communicate with the database server (MySQL) to query data, save changes, etc.
- MySQL is scalable and capable of handling more than 50 million rows.

Cons

- MySQL is not very efficient in handling very large databases.
- MySQL doesn't have as good a developing and debugging tool as compared to paid databases.
- MySQL versions less than 5.0 do not support COMMIT, stored procedure and ROLE.
- MySQL is prone to data corruption as it inefficient in handling transactions.
- MySQL does not support SQL check constraints.

Software used in this project:

Python will be used in this project for the backend. And for the front end, this project will use Flutter as a UI development kit to make an exemplary user interface. Python's syntax is simpler and more concise than Java and Kotlin as interpreted languages, and it can perform the same function as Java in fewer lines of code. As the application will not have a relational database for storing data, this project will use MongoDB, so the predefined commands and responses will be held in the MongoDB database as it uses JSON-like documents for storing data, which will be easier to access.

Chapter 3: Requirements and Analysis

3.1 Problem Definition:

As we have seen many voice assistants such as Google, Alexa, Bixby, and Cortana, the most common one in Windows is Cortana because it is embedded in the operating system. Cortana is a good voice assistant but lacks features such as voice activation, application closing, and shutting down the PC. Voice activation is simply saying the command to activate the voice assistant. So, in Cortana, we should first click the Cortana button, and then the voice activation happens by merely saying "Cortana", but only if the Cortana window is open. As a result, this is one of the reasons why users do not use it frequently. Without these constraints, the user environment could be more user-friendly. So, the VAW voice assistant project seeks to resolve these issues by incorporating the shortcomings of Cortana as features of this project and making the voice assistant more user-friendly.

3.2 Requirement Specifications:

VAW is a virtual assistant that uses speech recognition, natural language processing and speech synthesis to take actions to help its users. It will be an auto startup application, which means that when Windows is booted, it will also start in the background. So, a user can use the application right from the start of the windows just by saying a specific command, for example, "Hey VAW" or "Hello VAW" this feature can be called Voice Activated. As the VAW will work on the always listen to principle, it will detect the triggering command and get activated.

The other tasks performed by the VAW are creating a to-do list, setting a reminder, checking the weather, opening or closing the applications, playing music, Searching the web for information and taking notes by voice. The VAW can do these tasks on the command of the user.

3.3 Planning and Scheduling:

3.3.1 Planning:

Plan A

So, this project, VAW, is a desktop application. The user will click on the application icon to launch the application. Users can skip this step if VAW is already running in the background. Then the user says the command like "Hey VAW" or "Hello VAW," then the application gives a prompt on the screen "Listening," and a sound will be there. This prompt can be seen only when the user has opened the GUI of the application. If the GUI is not open in the foreground, users can only hear a prompt sound. This sound will indicate that VAW is in Listening mode. Now the user can say the command for the task performed by VAW. After completing the task, VAW will wait for the next order. If the order is not given, then VAW will go into standby mode.

Plan B

Plan A was about verbally saying the tasks to VAW. In plan b, the user can type in the job in the textbox of the VAW application, and it will do the required tasks for the user.

So, the user will open the application of VAW, click on the textbox present in the application, type in the task, and hit enter. VAW will do the job and give the results on the screen.

3.3.2 Scheduling:

Gannt Chart:

Below Gantt chart shows us the schedule of the project. It is a diagram representing the time taken for each task and can be easily prepared using Microsoft Word and Excel as well as some online applications. These types of charts help in the scheduling of the project after proper planning of the project is done. This scheduling helps to find out the actual time period of each stage of the project.

Task	Start Date	End Date	Duration
Chapter 1	08-Aug	14-Aug	7 Days
Chapter 2	26-Aug	05-Sep	11 Days
Chapter 3	06-Sep	14-Sep	9 Days
Chapter 4	15-Sep	25-Sep	10 Days

Figure 3.3.2.1 Gantt Chart (A)

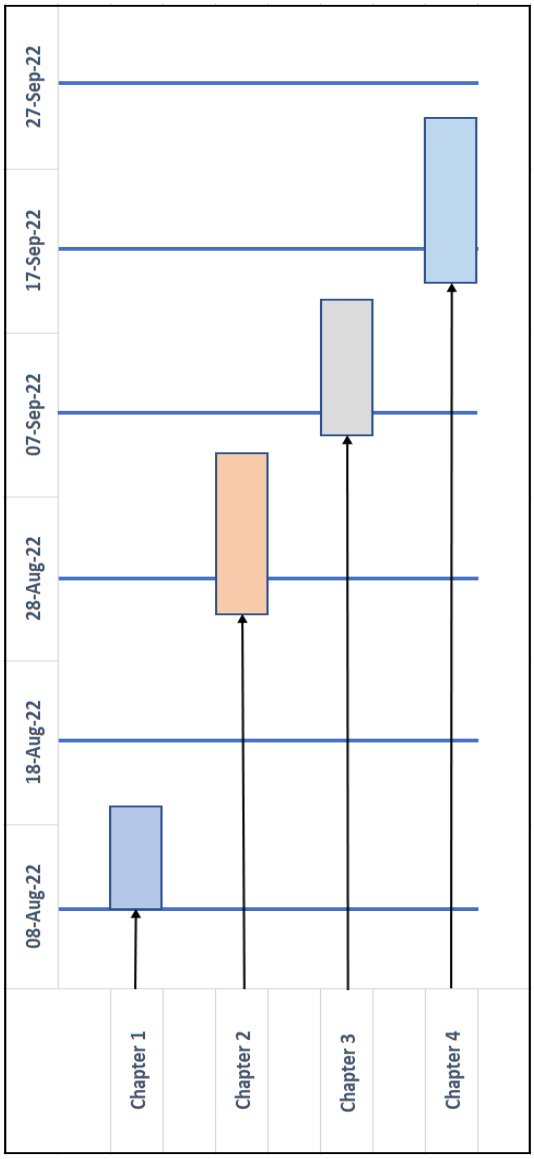
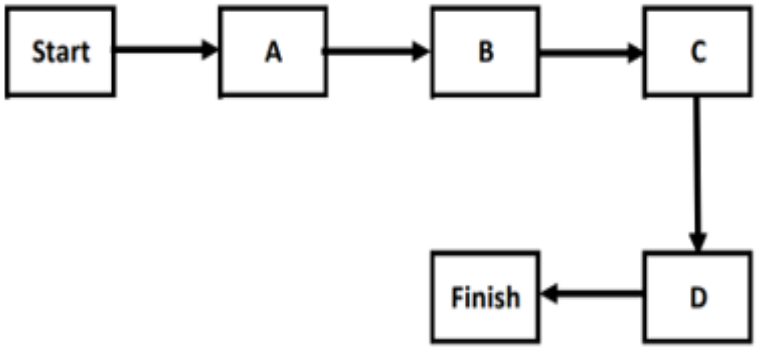


Figure 3.3.2.2 Gantt Chart (B)

CPM:

CPM or the Critical Path Method is an algorithm used in project management that is used to schedule project activities. The critical path refers to the longest stretch of the activities, and a measure of them from start to finish.

Chapter 1 = A
Chapter 2 = B
Chapter 3 = C
Chapter 4 = D



Activity	A	B	C	D
Predecessors	-	A	B	C
Duration	8	11	9	10

Figure 3.3.2.3 CPM (A)

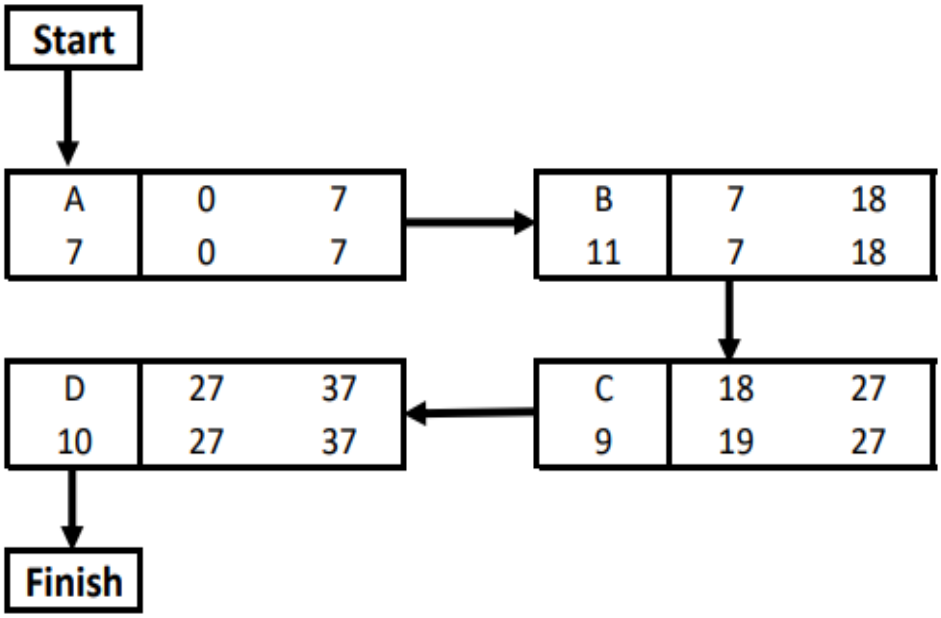


Figure 3.3.2.4 CPM (B)

3.4 Hardware and Software Requirement

3.4.1 Software Requirements

Python

PyCharm by JetBrains

MongoDB

Flutter

3.4.2 Hardware Requirements

Windows 7 (64-bit) or Higher

Minimum 8 GB RAM

Minimum 10 GB ROM

Core i3 4th Gen or Higher

3.5 Conceptual Model

Flow Chart

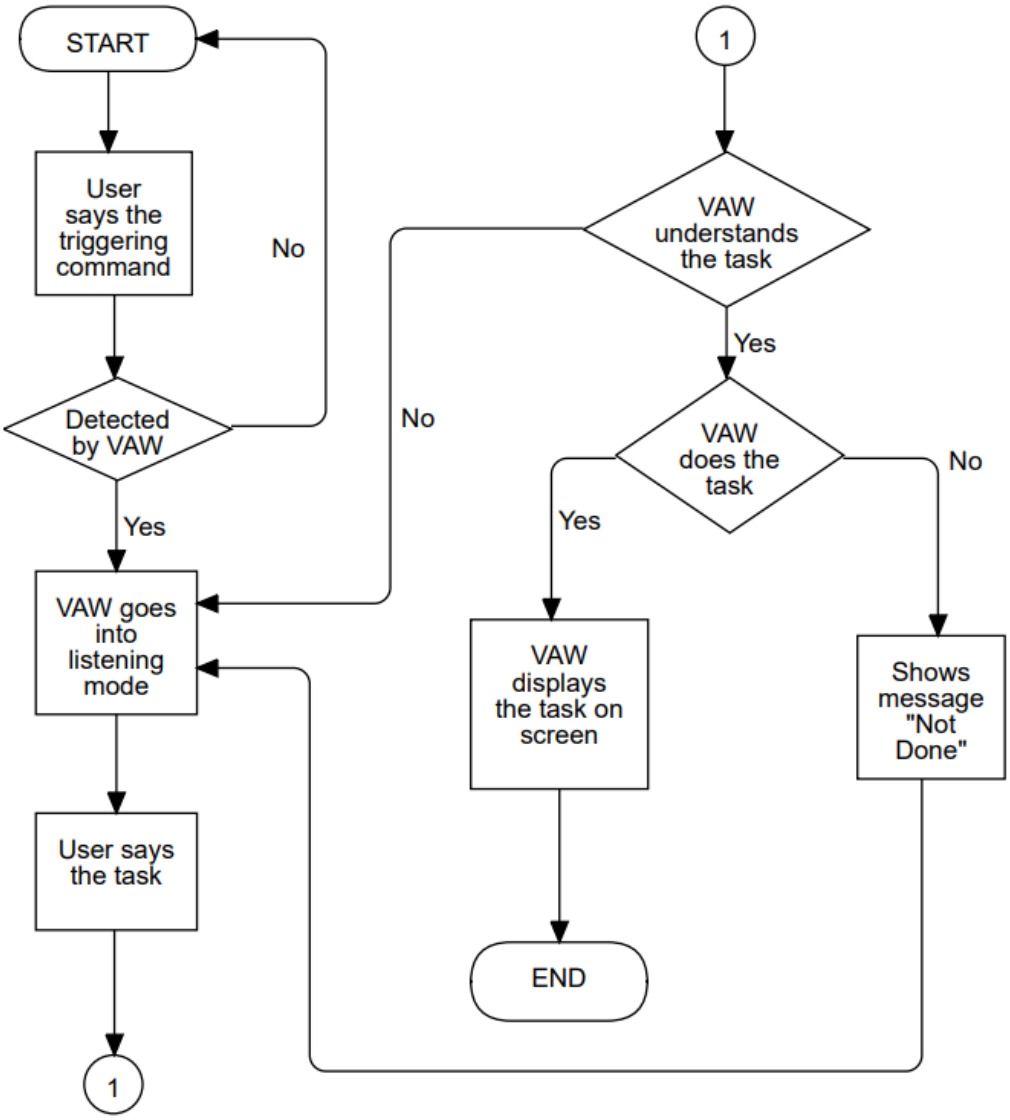


Figure 3.5.1 Flow Chart

Chapter 4: System Design

4.1 Basic Modules

The application will have the following modules.

- **GUI Module -** This module is for the User interface of the application. It will be a window with a mic button on the right side, and the text box and the messages displayed on the screen will be on the left side.

- **API Module** - This module will contain the API of different applications. So, the user can use the services of the various applications via these APIs.

- **Speech Recognition and Action Module** - This module will have the function to detect the voice/text of the user and perform specific actions for the given input by the user.

4.2 User Interface

The user interface will be a window with a mic button on the right side of the window, and the user can wake the voice assistant by using this button. And on the left of the window, the carried out the task and the user's input will be displayed in the form of a message, and there will be a textbox that the user will use to give the task to the application if the user prefers not to use the voice feature.

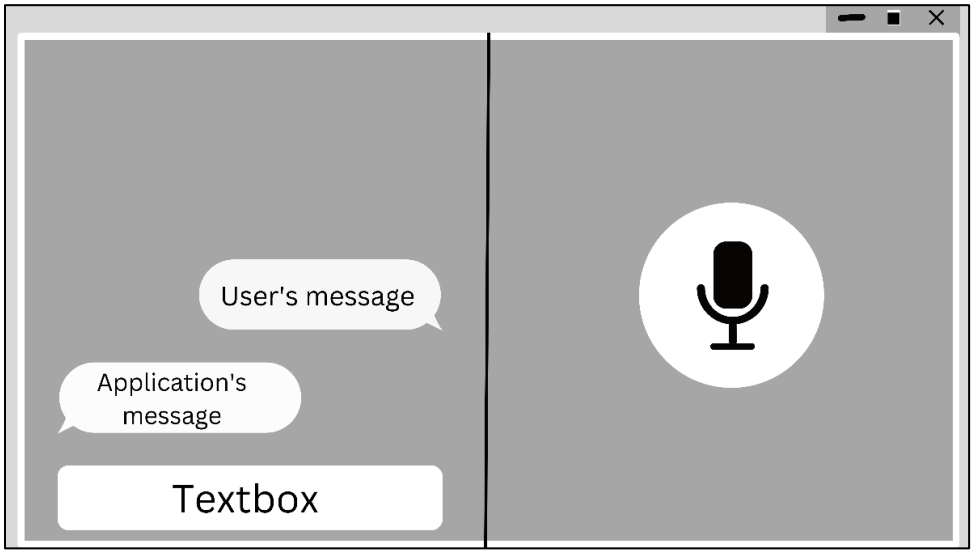


Figure 4.2.1 UI Design

4.3 System Design

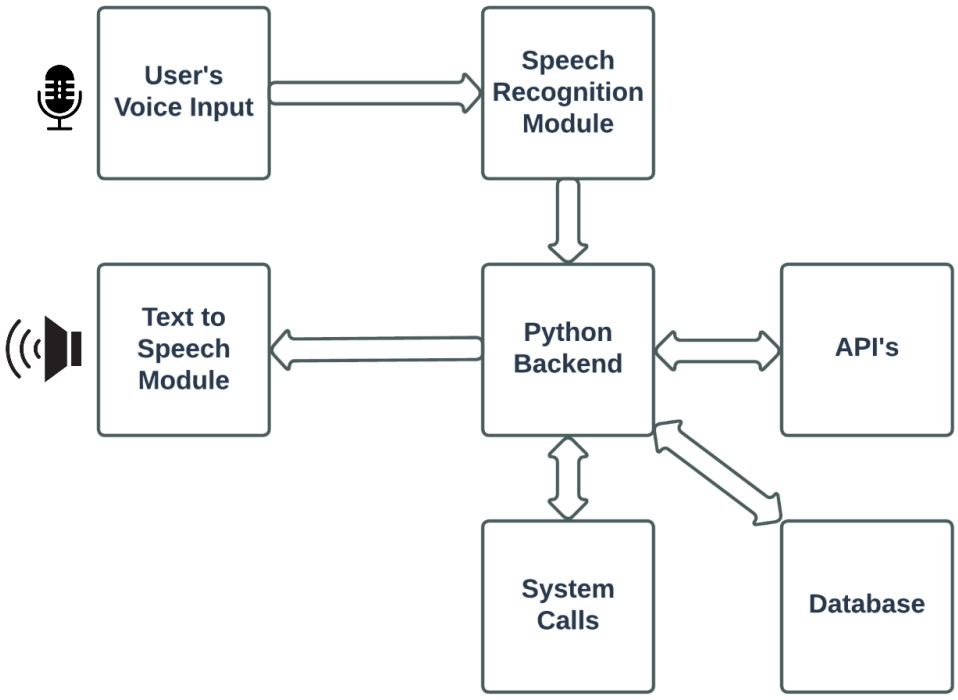


Figure 4.3.1 System Design

References

https://twitter.com/codemarch/status/1559774488392245248

https://www.codewithharry.com/

https://www.youtube.com/watch?v=9HaIGMyAyug&ab_channel=Hallden

SIES COLLEGE OF ARTS, SCIENCE & COMMERCE(AUTONOMOUS)
(Affiliated to University of Mumbai)
SION(W), MUMBAI-400022

DEPARTMENT OF INFORMATION TECHNOLOGY

SIES College of Arts, Science & Commerce
RISE WITH EDUCATION
Sion (West), Mumbai – 400022.
(Autonomous)

CERTIFICATE

This is to certify that the project entitled, "**VAW – Voice Assistant for Windows**", is bonafied work of **Mr. Jaikumar Gaja** bearing Seat. No: **TIT2223022** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Ms. Biju Ramesh
Internal Guide

Ms. Sudha B
Coordinator

External Examiner

Date: 23-03-2023

College Seal

Table of Content

Table of Figures

List of Tables

Changes made in Chapters 1, 2, 3 & 4

Software used in this project

The project will utilize the Python programming language. Python's syntax is comparatively more straightforward and compact when compared to Java and Kotlin, both of which are compiled languages. Despite being an interpreted language, Python can execute the same functionalities as Java, but with less code.

Gantt Chart:

Task	Start Date	End Date	Duration (Days)
Chapter 1	08-Aug	14-Aug	6
Chapter 2	26-Aug	05-Sep	10
Chapter 3	06-Sep	14-Sep	8
Chapter 4	15-Sep	25-Sep	10
Implementation	01-Nov	16-Feb	107
Chapter 5	17-Feb	20-Feb	3
Chapter 6	21-Feb	22-Feb	1
Chapter 7	23-Feb	24-Feb	1

Table 1 Gantt Chart

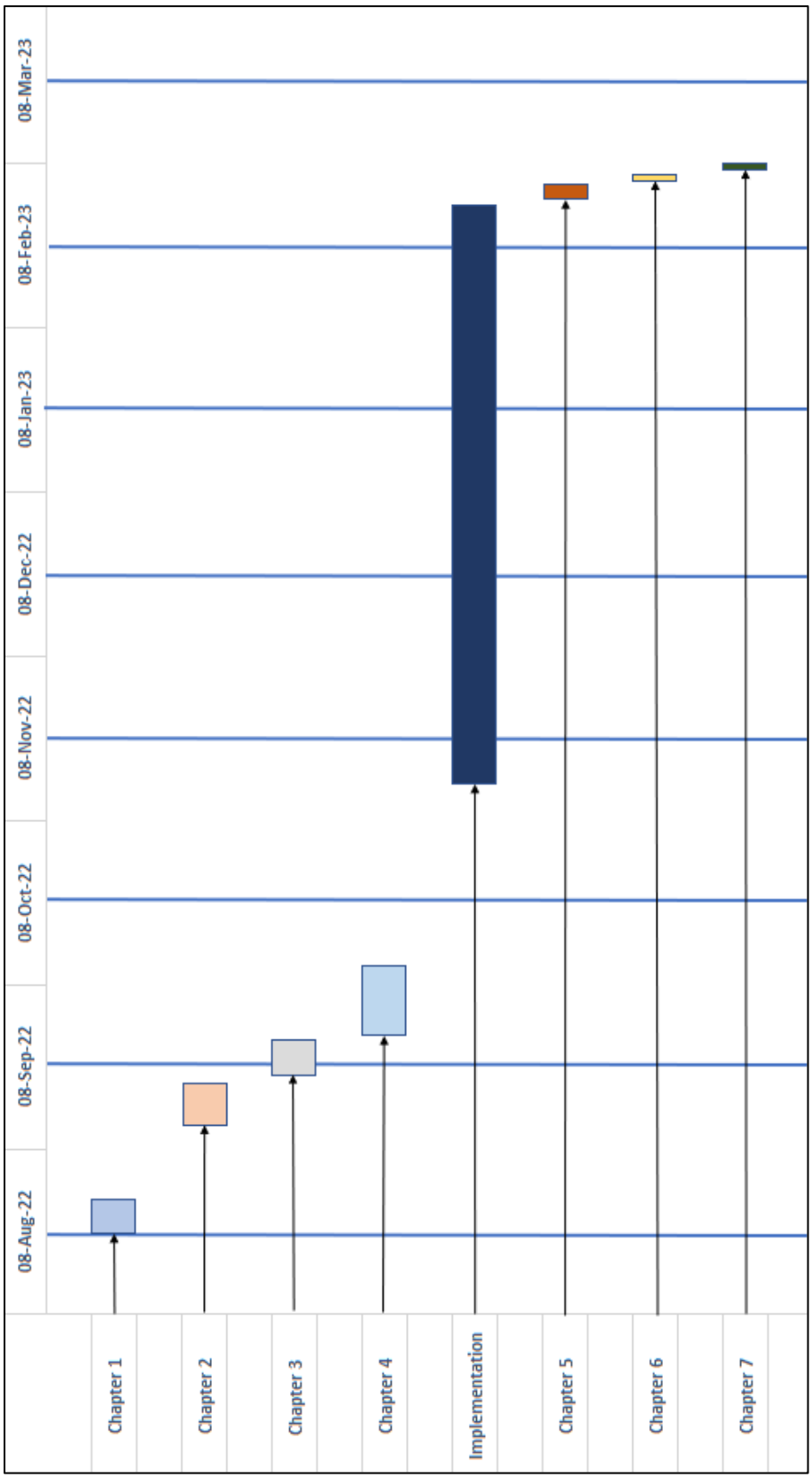
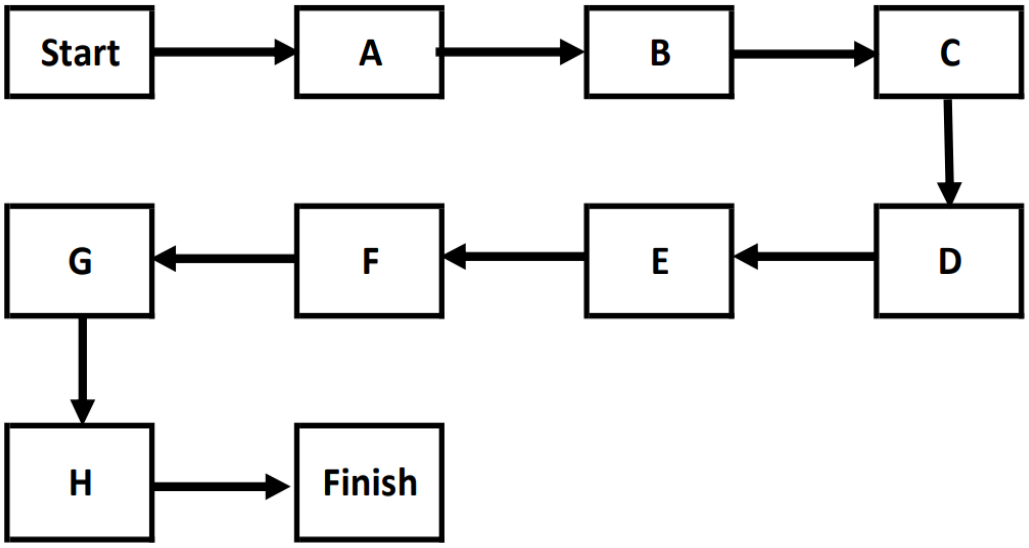


Figure 1 Gantt Chart

CPM:

Chapter 1 = A	Chapter 2 = B	Chapter 3 = C	Chapter 4 = D
Implementation = E	Chapter 5 = F	Chapter 6 = G	Chapter 7 = H



Activity	A	B	C	D	E	F	G	H
Predecessors	-	A	B	C	D	E	F	G
Duration	6	10	8	10	107	3	1	1

Table 2 Critical Activity

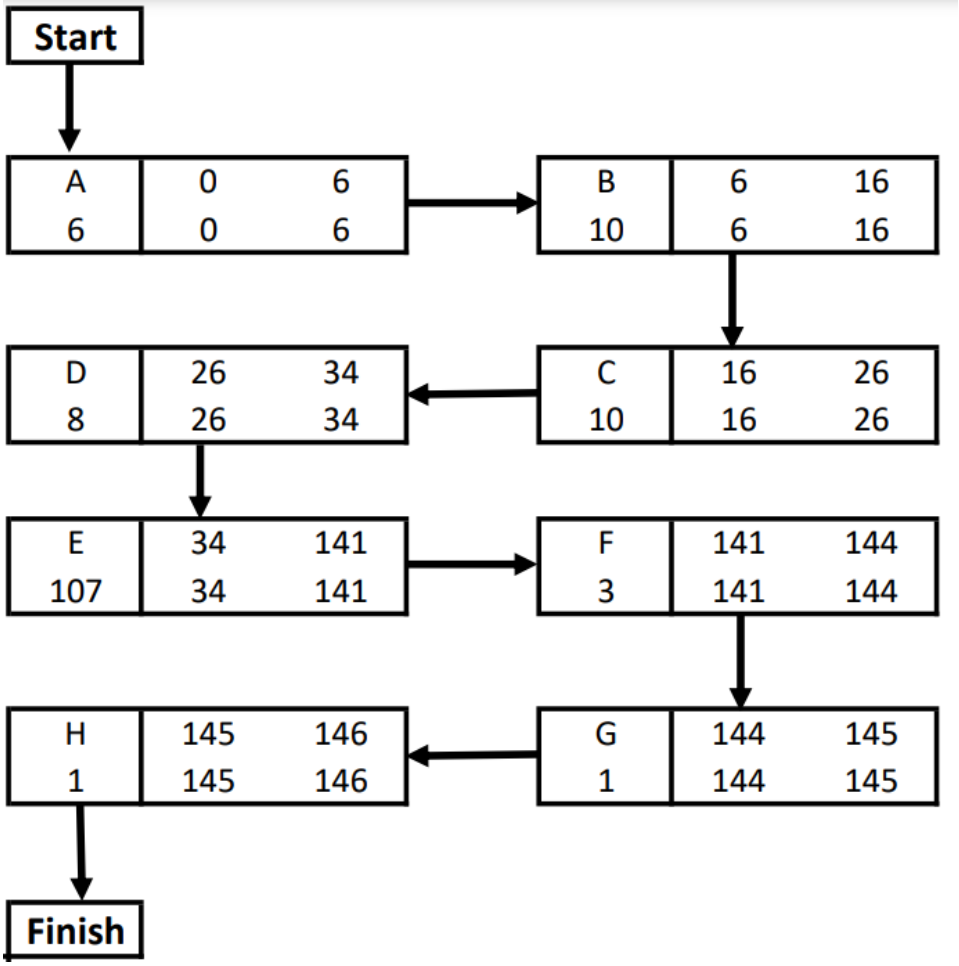


Table 3 Forward Backward Pass

Total Completion Time is 146. So, 146 Days

Slack = 0 for all the activities. Therefore, Critical Path = A-B-C-D-E-F-G-H

System Design:

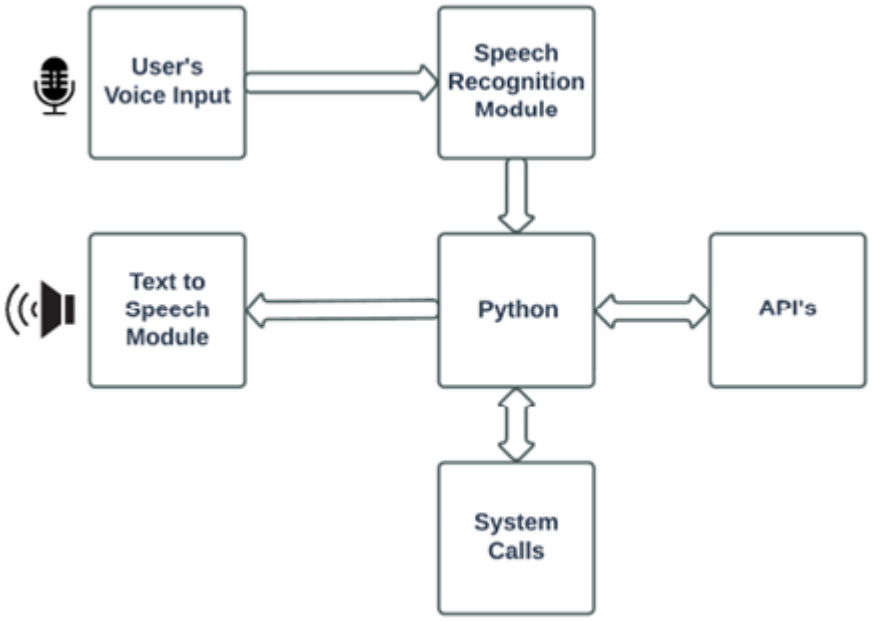


Figure 2 System Design

Chapter 5: Implementation and Testing

5.1 Implementation Approaches

The first step is to design a user-friendly graphical user interface (GUI). Once the GUI has been designed, the basic voice assistant code can be implemented using pre-built modules for speech recognition and text-to-speech. This code processes user's voice input and generates appropriate speech output.

Next, the GUI must be connected to the voice assistant code. This involves writing code to enable voice input and output through the interface. Finally, additional features and functionality can be added to the voice assistant using pre-built modules or custom code to make it more versatile and useful for users.

Upon launching the application, the user is greeted with a welcoming message. The voice assistant can be triggered by saying "hello Jarvis", which is followed by a confirmation response. Alternatively, the user can click the mic button on the graphical user interface (GUI) to skip the trigger command and directly give the assistant a command to perform. The GUI also includes a textbox that can be used in the same way as the mic button, allowing the user to enter commands by typing them.

5.2. Coding Details and Code Efficiency

5.2.1. Code

```
import os
import datetime
import tkinter
from tkinter import scrolledtext
import speech_recognition as sr
import wikipedia
import pyjokes
import webbrowser
import pyttsx3
import requests
import AppOpener
import threading
from tkinter import *

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)


class SpeakRecog:
    def __init__(self, scrollable_text):
        self.scrollable_text = scrollable_text

    engine = pyttsx3.init('sapi5')
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[1].id)

    def updating_ST(self, data):
        self.scrollable_text.configure(state='normal')
        self.scrollable_text.insert('end', data + '\n')
        self.scrollable_text.configure(state='disabled')
        self.scrollable_text.see('end')
        self.scrollable_text.update()

    def speak(self, audio):
        """It speaks the audio"""
        self.updating_ST(audio)
        self.engine.say(audio)
        self.engine.runAndWait()

    def print(self, data):
        self.updating_ST(data)


def takeCommand():
    r = sr.Recognizer()

    with sr.Microphone() as source:
```

```
        SR.print("Listening...")
        r.pause_threshold = 2
        audio = r.listen(source)
    try:
        SR.print("Recognizing...")
        queryString = r.recognize_google(audio,
language='en-in')
        SR.print(f"User said: {queryString}\n")
    except Exception as e:
        # print(e)
        SR.print("Unable to Recognize your voice.")
        return "None"
    return queryString


def weather(queryString):
    if queryString.startswith('weather of'):
        li = list(queryString.split(" "))
        loc_base =
"http://api.openweathermap.org/geo/1.0/direct?q="
        apikey = "9bf14a035688f500531aa388ffefac84"
        city = li[2]
        loc_url = loc_base + city + "&appid=" + apikey
        response = requests.get(loc_url).json()
        lat = response[0]["lat"]
        lon = response[0]["lon"]
        base =
"https://api.openweathermap.org/data/2.5/weather?lat="
        url1 = base + str(lat) + "&lon=" + str(lon) +
"&appid=" + apikey + "&units=metric"
        response1 = requests.get(url1).json()
        temp = response1['main']['temp']
        feels_like = response1['main']['feels_like']
        humidity = response1['main']['humidity']
        description =
response1['weather'][0]['description']
        SR.speak(f'The Temperature is {temp}')
        SR.speak(f'It Feels Like {feels_like}')
        SR.speak(f'The Humidity is {humidity}')
        SR.speak(f'It is {description} outside')
    else:
        SR.speak(f'Please Say., Weather. of. city name ')


def wishMe():
    hour = int(datetime.datetime.now().hour)
    if 0 <= hour < 12:
        SR.speak("Good Morning Sir !")

    elif 12 <= hour < 18:
        SR.speak("Good Afternoon Sir !")
```

```
    else:
        SR.speak("Good Evening Sir !")


def wikipediaFn(queryString):
    SR.speak('Searching Wikipedia...')
    qs = queryString.replace("wikipedia", "")
    results = wikipedia.summary(qs, sentences=3)
    SR.speak("According to Wikipedia")
    SR.print(results)
    SR.speak(results)


def youtube(queryString):
    SR.speak("Here you go to Youtube\n")
    webbrowser.open("youtube.com")


def google(queryString):
    SR.speak("Here you go to Google\n")
    webbrowser.open("google.com")


def stackoverflow(queryString):
    SR.speak("Here you go to Stack Over flow.Happy coding")
    webbrowser.open("stackoverflow.com")


def appOpen(queryString):
    appname = list(queryString.split(" "))
    AppOpener.run(appname[1])
    SR.speak("opening")


def CommandsList():
    os.startfile('Commands.txt')


def search(queryString):
    if 'search' in queryString:
        qs = queryString.replace("search", "")
    else:
        qs = queryString.replace("play", "")
    webbrowser.open(qs)


def commands(queryString):
    if 'wikipedia' in queryString:
        wikipediaFn(queryString)

    elif 'open youtube' in queryString:
        youtube(queryString)
```

```
elif 'open google' in queryString:
    google(queryString)

elif 'open stack overflow' in queryString:
    stackoverflow(queryString)

elif 'how are you' in queryString:
    SR.speak("I am fine, Thank you")
    SR.speak("How are you, Sir")

elif 'fine' in queryString or "good" in queryString:
    SR.speak("It's good to know that your fine")

elif 'joke' in queryString:
    SR.speak(pyjokes.get_joke())

elif 'exit' in queryString:
    SR.print("User said : Exit")
    SR.print("Thanks for giving me your time")
    SR.speak("Thanks for giving me your time")
    exit()

elif 'shutdown' in queryString:
    SR.speak("Shutting down PC")
    os.system("shutdown /s /t 1")

elif 'restart' in queryString:
    SR.speak("Restarting PC")
    os.system("shutdown /r /t 1")

elif 'weather' in queryString:
    weather(queryString)

elif 'open' in queryString:
    SR.print(queryString)
    appOpen(queryString)

elif 'search' in queryString:
    search(queryString)

elif "write a note" in queryString:
    SR.speak("What should i write, sir")
    note = takeCommand().lower()
    file = open('VAW.txt', 'w')
    SR.speak("Noted")
    file.write(note)

elif "show the note" in queryString:
    SR.speak("Showing Notes")
    file = open("jarvis.txt", "r")
    # print(file.read())
```

```
        SR.speak(file.read())


def mainEntry(event):
    def clearScr():
        var = lambda: os.system('cls')

    clearScr()
    wishMe()

    while True:
        queryString = takeCommand().lower()
        if 'hello jarvis' in queryString:
            SR.speak('Yes Sir,')
            while True:
                queryString = takeCommand().lower()
                commands(queryString)
                break


def formicbtn():
    event.clear()
    SR.print("Done")
    queryString = takeCommand().lower()
    commands(queryString)
    event.set()
    SR.print("Set")


event = threading.Event()
mainThread = threading.Thread(target=mainEntry,
args=(event,))

if __name__ == "__main__":
    window = Tk()
    window.geometry("900x500")
    window.configure(bg="#FFFFFF")
    window.resizable(False, False)

    enterImg = PhotoImage(file='EnterButton.png')
    micImg = PhotoImage(file='MicButton.png')
    partition = Canvas(window, bg="#FFFFFF", height=500,
width=900, bd=0, highlightthickness=0, relief="ridge")
    partition.place(x=0, y=0)
    partition.create_rectangle(449.0, -1.0, 452.0, 500.0,
fill="#000000", outline="")

    textbox = Entry(window, border=2)
    textbox.place(x=14.0, y=430.0, width=353.0,
height=53.0)

    scrollable_text = scrolledtext.ScrolledText(window,
```

```
state='disabled', relief='sunken', bd=5, wrap=tkinter.WORD,

bg='#add8e6', fg='#800000')
    scrollable_text.place(x=14.0, y=13.0, height=410.0,
width=425.0)
    SR = SpeakRecog(scrollable_text)


    def enterClick():
        keyboardString = textbox.get()
        queryString = keyboardString
        SR.print(keyboardString)
        commands(queryString)


    myButton1 = Button(window, image=enterImg,
borderwidth=0, command=enterClick)
    myButton1.place(x=384.0, y=430.0, width=55.0,
height=55.0)

    myButton2 = Button(window, image=micImg, borderwidth=0,
bg='white', command=formicbtn)
    myButton2.place(x=638.0, y=193.0, width=115.0,
height=115.0)


    def closing():
        exit()


    myMenu = tkinter.Menu(window)
    m1 = tkinter.Menu(myMenu, tearoff=0)
    m1.add_command(label='Commands List',
command=CommandsList)
    myMenu.add_cascade(label="Help", menu=m1)

    window.config(menu=myMenu)

    mainThread.start()
    window.mainloop()
```

5.2.2. COCOMO

The COCOMO model provides a framework for estimating project costs and resource requirements, which can be useful for project planning and management. However, it's important to note that the accuracy of the estimate depends on the accuracy of the input parameters and the complexity of the project itself.

- **Organic**: This variant is used for small software projects with a team of experienced developers. The development process is typically informal, and the project requirements are well understood. The Organic variant is characterized by a relatively small team size, a low level of complexity, and a high degree of familiarity with the development tools and techniques.

- **Semi-detached**: This variant is used for medium-sized projects with a mix of experienced and inexperienced developers. The development process is more formal than the Organic variant, and the project requirements may be less well understood. The Semi-detached variant is characterized by a moderate team size, moderate complexity, and a mix of familiar and unfamiliar development tools and techniques.

- **Embedded**: This variant is used for large-scale software projects with a large team of developers. The development process is highly formal and structured, and the project requirements may be highly complex and poorly understood. The Embedded variant is characterized by a large team size, high complexity, and a mix of familiar and unfamiliar development tools and techniques.

The formula used for the cost estimation for the COCOMO model is given below:

$Effort = a*(KLOC)^b$

[KLOC=lines of code(thousands)]

$Time = c*(Effort)^d$

Staffing = Effort/Time

Type	a	b	c	d	KLOC
Organic	2.4	1.05	2.5	0.38	0.718
Effort (Person-months)			1.69		
Time (months)			3.06		
Staffing			0.55		

Table 4 COCOMO

5.3. Testing Approaches

Testing is a process used to verify that the software product aligns with the expected requirements and is free of defects. It entails running the software or system components with the aid of manual or automated tools in order to assess one or more properties of interest.

Manual testing is a type of software testing that involves having human testers execute test cases on the voice assistant. This is done to ensure that the voice assistant functions correctly and meets the requirements of the user. During manual testing, testers can simulate different voice commands and check the responses of the assistant. It is a good option for a simple voice assistant because it can help identify any issues that may arise during actual usage.

5.4. Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
1	Launching the voice assistant	Say "Hello Jarvis" or “Hey Jarvis”	The voice assistant should activate and greet the user	Voice assistant said “yes Sir”	PASS
2	Command execution	Say "Weather of Mumbai"	The voice assistant should respond with the current weather forecast	Voice assistant responded with the current weather of Mumbai	PASS

3	Command execution	Say "Open YouTube"	The voice assistant should open YouTube page in browser	Opened YouTube Page	PASS
4	Command execution	Say "Open Google"	The voice assistant should open google search page in browser	Opened Google Page	PASS
5	Command execution	Say "Open Stack Overflow"	The voice assistant should open stack overflow page in browser	Opened Stack overflow page	PASS
6	Command execution	Say "Tell me a joke"	The voice assistant should tell you a joke	Told a joke	PASS
7	Command execution	Say "shutdown PC"	The voice assistant should shut down the PC	Turned off the PC	PASS
8	Command execution	Say "restart the PC"	The voice assistant should Restart the PC	Restarted the PC	PASS
9	Command execution	Say "Open Calendar"	The voice assistant should open calendar application	Opened Calendar	PASS
10	Command execution	Say "Open Control Panel"	The voice assistant should open	Opened Control panel	PASS

			control panel		
11	Command execution	Say "Open Discord"	The voice assistant should open discord	Opened Discord	PASS
12	Command execution	Say "Search Close eyes music"	The voice assistant should open browser and search for close eyes	Searched Close eyes	PASS
13	Command execution	Say "Play Middle of the night"	The voice assistant should open browser and play	Opened YouTube and played Middle of the night	PASS
14	Command execution	Say "How much butter is enough"	The voice assistant should open the browser	Open the browser and searched for How much butter is enough	PASS
15	Command execution	Say "Write a Note"	The voice assistant should respond with "What should I write sir".	Responded	PASS
16	Command execution	Say "Show the note"	The voice assistant should show the note	Showed the note	PASS
17	Basic Window Operation	Click on Close Window Button	The application should be closed	Application closed	PASS
18	Basic Window Operation	Click on Minimize button	The application	Application minimized	PASS

			should be minimized		
19	Application Function	Click on Mic Button	you should hear a sound	heard the sound	PASS
20	Application Function	Click on Text Box	Cursor should appear	Cursor appeared on the text box	PASS
21	Application Function	Type something after click on the text box	The Typed words should be visible	Typed words visible in the text box	PASS
22	Application Function	Click the Button next to Text box	The typed words should be visible on the application and should perform the task	Typed words visible and performed the task	PASS

Table 5 Test Case

Chapter 6: Results and Discussions

6.1. Test Report

A test report is a comprehensive document that summarizes all test activities and final test results of a software testing project. It provides an assessment of how well the testing was conducted and identifies any issues that were encountered. The primary objective of testing is to verify if various features within an application are functioning as per the specified requirements, and the test report helps to ensure that the expected results are achieved.

For testing the Voice Assistant, a computer with an Intel Core i3 4th Gen processor and 8 GB of RAM was used, running on the Windows 10 version 21H2 operating system. The PyCharm software was used to run the Voice Assistant application.

To perform testing on the voice assistant based on the defined test cases, manual testing was conducted. Since the voice assistant requires user interaction, it was necessary to manually simulate different voice commands to ensure that the assistant functions correctly. During testing, it was observed that the application froze when the mic button was clicked. However, upon further investigation, it was discovered that the application was still running in the background and detecting the voice input. To address this issue, a separate thread was created to run the function responsible for processing the voice input when clicking the mic button.

6.2 User Documentation

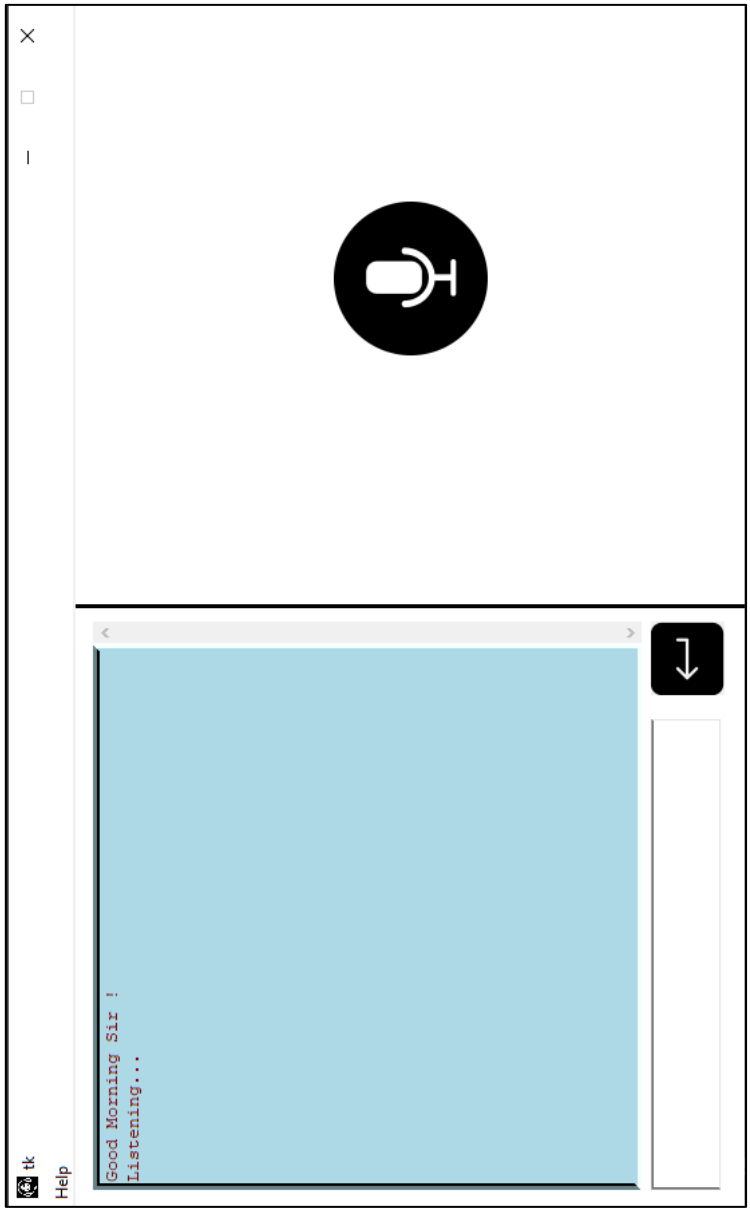


Figure 3 Launching Application

After launching the application, the Voice assistant greets with a message.

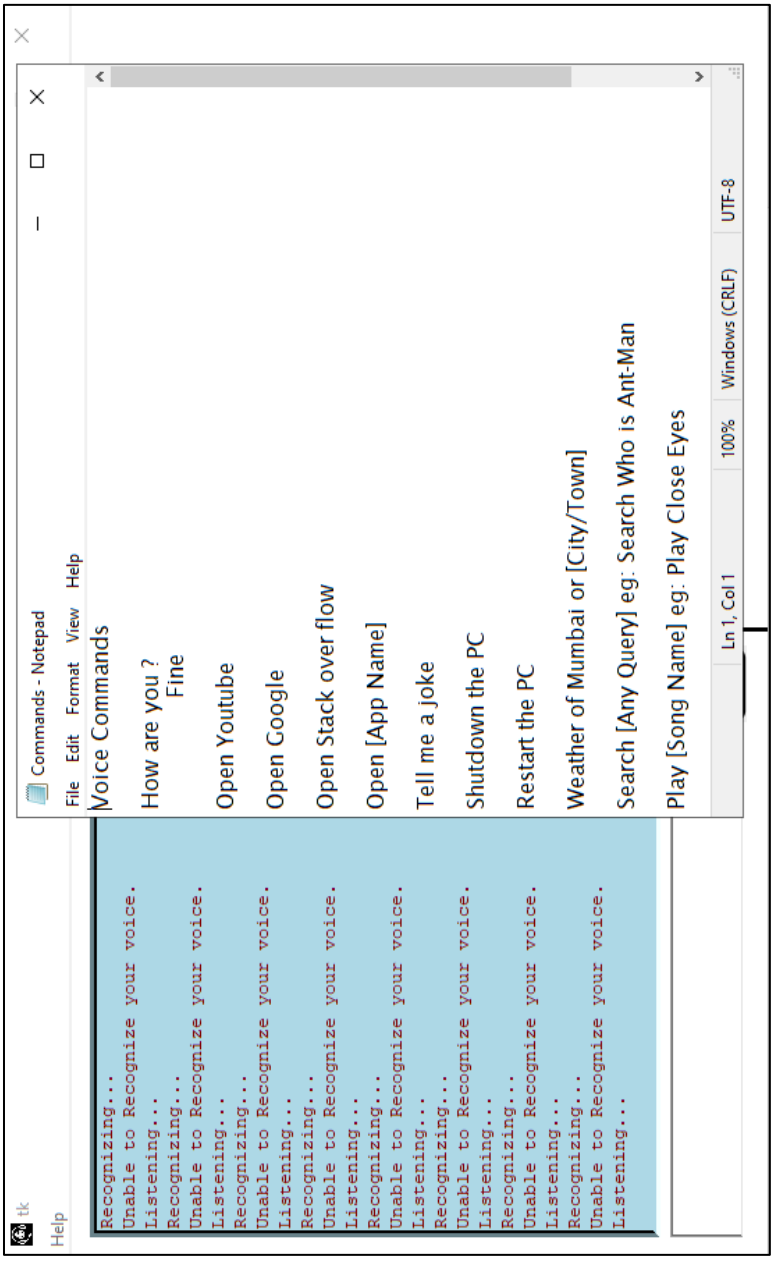


Figure 4 Help Button

When a user click help and the click Command List a text file should appear with commands

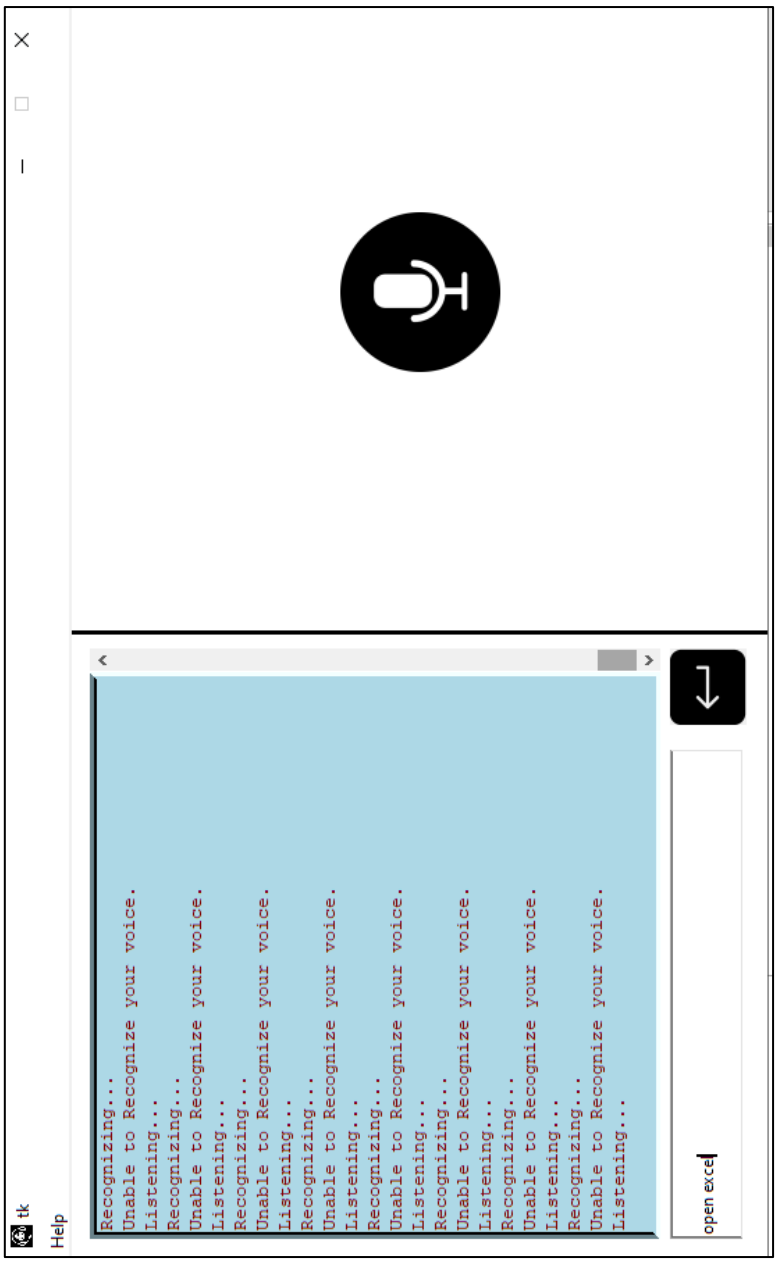


Figure 5 Text Box

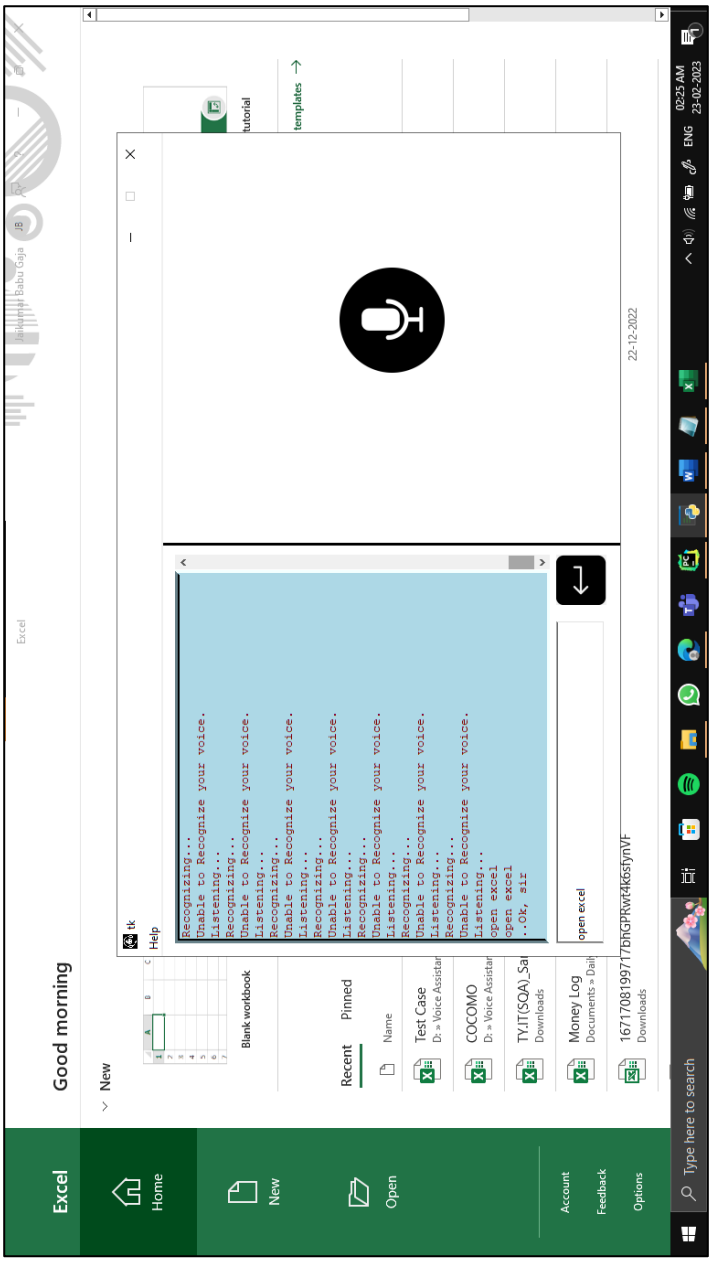


Figure 6 Text Box Working

User writes the command in the text box, then the command appeared on the application and performed the task.

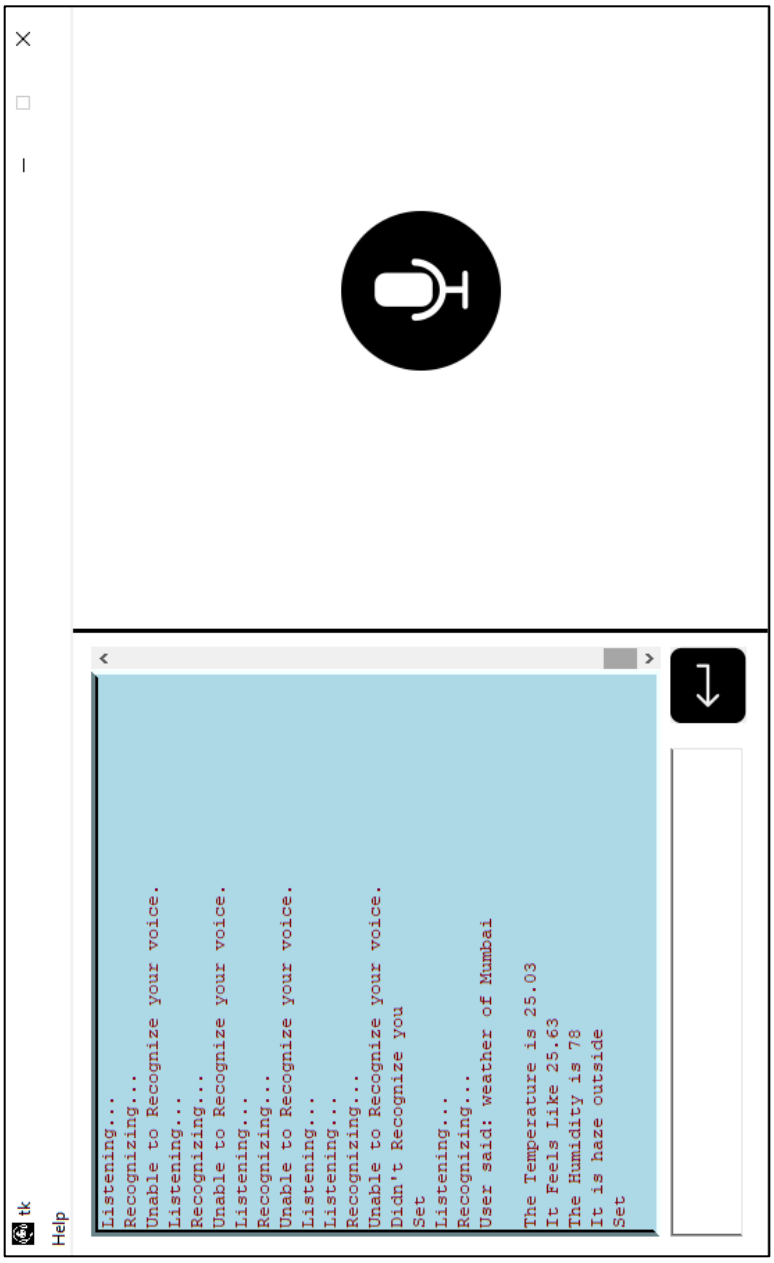


Figure 7 Weather Command

The user clicked the mic button and said “weather of Mumbai” so the weather description seen on the application.

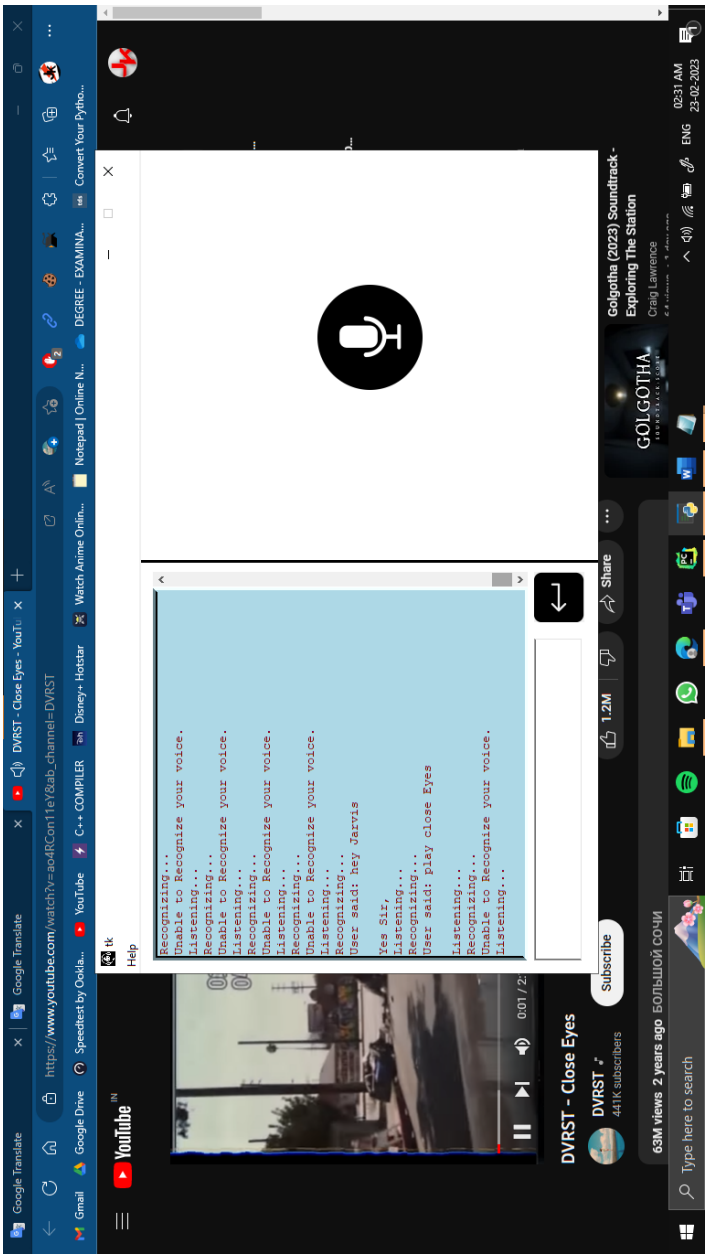


Figure 8 Playing Music

The user said “Hey Jarvis” then the assistant said “yes sir” then the user said “play close eyes” the assistant performed the task

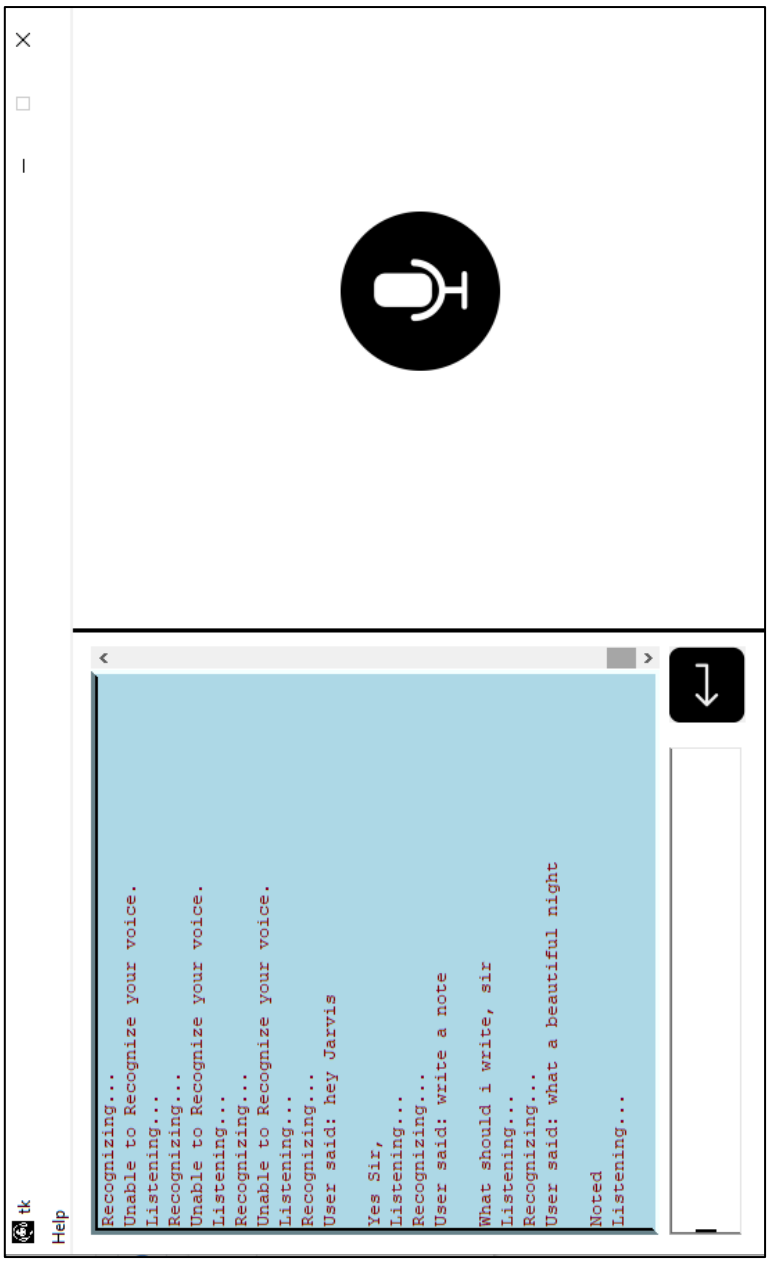


Figure 9 Write Note Command

The user writing a note through command.

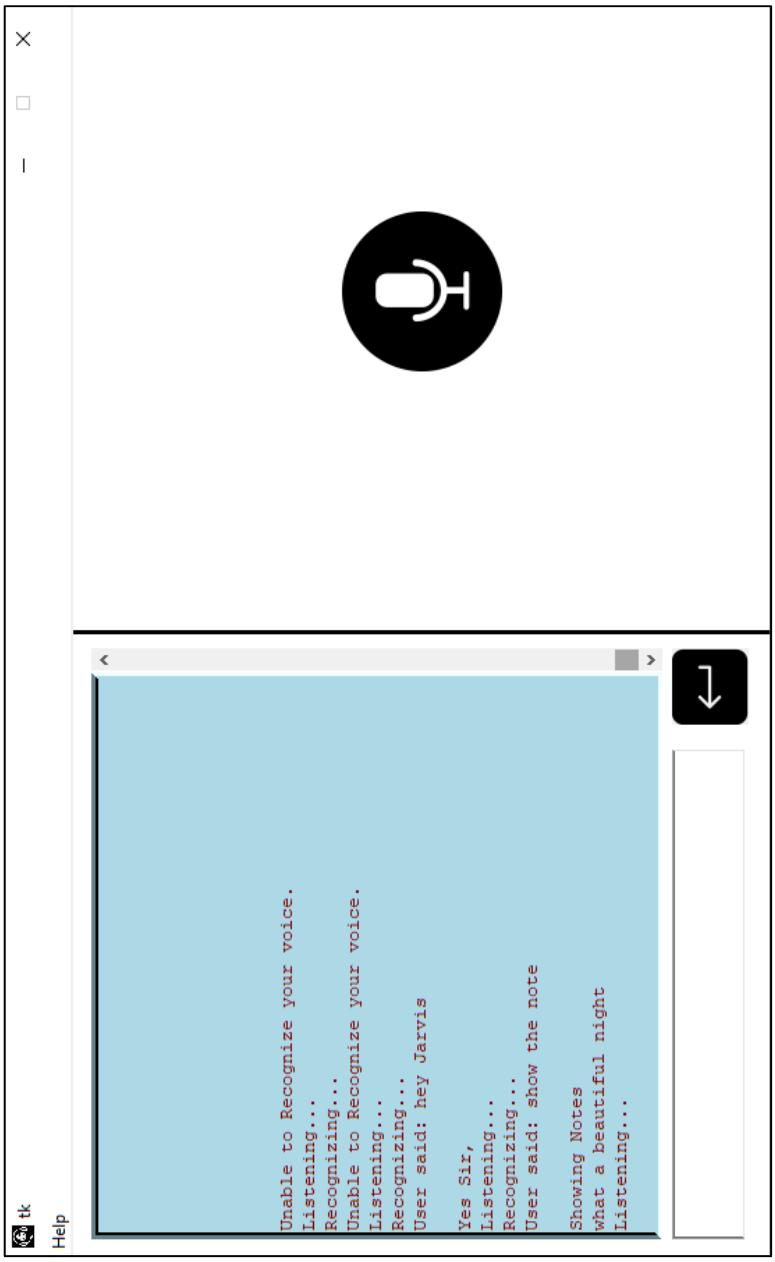


Figure 10 Show Note Command

Voice assistant showing the note.

Chapter 7: Conclusion

7.1 Significance of the system:

Voice assistant can make certain tasks more convenient by allowing the user to perform them using voice commands. For example, a user can use a voice assistant to search for something, open applications, play music, or check the weather without having to physically interact with a device.

Voice assistant can make technology more accessible for people who have difficulty using traditional interfaces. For example, a voice assistant can be helpful for people with visual or motor impairments.

Voice assistant can improve efficiency by allowing the user to perform tasks more quickly. For example, a user can use a voice assistant to play music while doing other things at the same time.

7.2 Limitations of the system:

One limitation of voice assistant is that they cannot dynamically adapt to every variation of a user's voice command. This means that the user needs to use specific commands or phrases that are programmed into the system to get the desired action to be performed. If the user deviates too much from the specific command or phrase, the voice assistant may not recognize it and may not perform the desired action. This can limit the usefulness of the voice assistant and require the user to remember specific commands or phrases in order to use the system effectively.

Limited commands: This can limit the functionality of the voice assistant and user require the user to memorize specific commands or phrases in order to perform actions. It also means that the assistant may not be able to adapt to new or unique commands from the user, which can be frustrating for some users.

7.3 Future Scope of the Project:

The project meets all the specified requirements, but there is potential to enhance it with additional features.

Creation of Exe File: In the current version of the project, the generated exe file is not completely functional due to errors in some modules. However, we can work on resolving these errors in the future and create a standalone exe file for the voice assistant.

Using of AI: AI can be used to train a voice assistant to provide responses to a wide range of user queries. This functionality can be incorporated into the voice assistant to improve its performance and accuracy.

References

Vishal Patil - YouTube

Stack Overflow - Where Developers Learn, Share, & Build Careers

The fastest way to build Flutter apps in Python | Flet

Qt for Python

ChatGPT: Optimizing Language Models for Dialogue (openai.com)

Documentation (python.org)

Figma: the collaborative interface design tool.

Avi Upadhyay - YouTube