PROJECT REPORT ON

Voice Assistant and Notepad

Submitted in partial fulfillment of the requirements for the

degree of

Bachelors of Science in Computer Science

Submitted by

Debargha Nandi(Roll no:**203513-21-0110** Reg No:**513-1114-0280-20)**

Sk. Manjarul Hossain(Roll No:**203513-21-0017** Reg No:**513-1111-0315-20)**

Pinaki Nandan Parya(Roll No:**203513-21-0043** Reg No:**513-1111-0401-20)**

Under the Guidance of

Mr. Debasish Kundu

(SACT)

Department of Computer Science, Sammilani Mahavidyalaya

Under

University of Calcutta

DECLARATION

We hereby declare that this project is based on our original work except or citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at any university or any other institutions.

Debargha Nandi

Roll no:**203513-21-0110**

Reg No:**513-1114-0280-20**

Sk. Manjarul Hossain

Roll No:**203513-21-0017**

Reg No:**513-1111-0315-20**

Pinaki Nandan Parya

Roll No:**203513-21-0043**

Reg No:**513-1111-0401-20**

Department of Computer Science, Sammilani Mahavidyala

Under

University of Calcutta

Certificate of Approval

This is to certify that the dissertation is the record of Final Year Project, entitled **“Voice Assistant and Notepad”** carried out by **Debargha Nandi (**Roll No:**203513-21-0110** and Reg No:**513-1114-0280-20**), **Sk Manjarul Hossain** (Roll No:**203513-21-0017** and Reg No:**513-1111-0315-20**) and **Pinaki Nandan Parya** (Roll No:**203513-21-0043** and Reg No:**513-1111-0401-20**) of the department of Computer Science, Sammilani Mahavidyalaya for the partial fulfillment of the award of the degree of Bachelor of Science (Session 2022-2023) by University of Calcutta in the year 2023 under my supervision and guidance. To the best of my knowledge, the results embodied in this report , are original in nature and worthy of incorporation in the present version of the report for B.Sc. programme in Computer Science.

This report has not been submitted to any other university or institution for the award.

Head of the department	Guide/Supervisor
______________________	______________________
Mrs. Brototi Mondal	Mr. Debasish Kundu
(Assistant Professor)	(SACT)

ACKNOWLEDGEMENT

First of all, we, **Debargha Nandi, Sk Manjarul Hossain** and **Pinaki Nandan Parya** of B.Sc. Computer Science Semester VI would like to express our profound sense of gratitude towards our Project Guide **Mr. Debasish Kundu,** SACT, Department of Computer Science, for his able guidance, support and encouragement during the course of my Semester VI project: **Voice Assistant and Notepad.**

I am deeply indebted to our Project Guide for giving us this opportunity to work on this project and for his kind help and support to develop an understanding of the subject and making a clear knowledge by providing necessary insight. His readiness for consultation at all times, his educative comments, his concern, concrete support and assistance even with practical things have been invaluable. Lastly, I would like to thank the entire faculty of Computer Science Department of our college for cultivating a healthy and creative environment to work in the project.

Debargha Nandi
Roll No:**203513-21-0110**
Reg No:**513-1114-0280-20**

Sk Manjarul Hossain
Roll No:**203513-21-0017**
Reg No:**513-1111-0315-20**

Pinaki Nandan Parya
Roll No:**203513-21-0043**
Reg No:**513-1111-0401-20**

Date:

Place: Kolkata

Abstract

The purpose of this project, **Voice Assistant and Notepad** is to help students note down the lectures given by their respected teachers or professors using the Voice Notepad option. The additional feature of the Voice Assistant and Notepad developed by us, is that it comes with the Voice assistant feature where you can give command to your machine to open or run things on your machine. This will ensure that you get all the lectures noted down accurately in your machine and can save it to use it later. And the Voice Assistant will help you perform tasks without putting your hands on the keyboard. The Voice Notepad can accurately recognize Hindi and English terms. A separate GUI is provided for the Voice Notepad where there are 3 options: Export, Speak and Reset.

Finally, this project was concluded with the scope for further work which can be done to achieve better results.

TABLE OF CONTENTS

Chapter 1: INTRODUCTION……………………………………………………………6
1.1 Introduction…………………………………………………………………...6
1.2 Motivation……………………………………………………………………6
1.3 Scope & Background of the work…………………………………………………7
1.4 Objective………………………………………………………………………8

Chapter 2: METHODOLOGY……………………………………………………………9
2.1 Introduction to Methodology…………………………………………………9
2.2 Project Structure…………………………………………………………9-10
2.3 Algorithm……………………………………………………………………..11

Chapter 3: IMPLEMENTATION………………………………………………………..12
3.1 About Voice Assistant and Notepad………………………………………….12-13
3.2 Modules Used……………………………………………………………..13-16
3.3 Language Used…………………………………………………………16-17
3.4 Resources Used for this Project…………………………………………………17

Chapter 4: ANNEXURE(SOURCE CODE)………………………………………18-63

Chapter 5: Result and Discussion……………………………………………………...64
5.1 Result and Discussion……………………………………………………..64-65
5.2 Limitations of this project…………………………………………………….66
5.3 Future Work……………………………………………………………….66-67

Chapter 6: CONCLUSION…………………………………………………………….68

REFERENCES……………………………………………………………………….69

Chapter 1: INTRODUCTION

1.1 Introduction

A Voice Assistant and Notepad can be defined as an application for students to manage taking notes in the class and manage their computers without moving their hand as well. It is an area that is giving detailed information about Speech Recognition System and its operation and hidden talent. Capabilities of this project include Voice Notepad and Voice Assistant, as well as storing what is said to the Voice Notepad. Majority of the Voice Assistant available till date comes with only the Assistant thing which will follow some orders given by the user. But we are adding the Voice Notepad feature in it to help everyone who wants to keep record of what is said in front of them. In the market there is either Voice Assistant or Voice Notepad (in limited numbers). But we are combining this two in this project. We are committed to continuously improving Voice Assistant and Notepad by leveraging the latest advancements in artificial intelligence. This means you can expect regular updates and enhancements to further enhance your speech recognition experience.

1.2 Motivation

Speech Recognition is already a part of our daily life like searching for news, turning on or off applications, asking assistant to tell a joke, but for now it is still limited to relatively simple commands. As technology advances, researchers will be able to make more intelligent system that will understand conventional speech. One day you will able to talk to your computer the way you talk to any other human being, and it will be able to respond to your words. All this will be made possible with the technologies like **Voice Assistant and Notepad.** The number of specialists required in this field is growing and many companies are looking for talented people who want to be a part of it. As per current upcoming needs speech recognition will be a fast growing and world changing subset of signal processing for years to come.

1.3 Scope & Background of the work

The **Voice Assistant and Notepad** is an application that gives opportunities to students. This system manages in individual interfaces and dictation paired with transcriptions costs and a much easier workflow. Voice recognition system can be used by any industry. Work from anywhere using a secured internet connection. The range of this application is very wide and diversified. The intentions are to reduce complexity in the world of education and students. Besides this it can be accessed by other people too who needs to note down or perform tasks etc. In addition to being precise and accurate, Speech Recognition technology can detect accents and spell words accurately.

1.3.1 Background of The Work

A review of literature is the section of any research study which provides a critical view and detailed overview of all the magnitudes of the specific subject of study, which has already been exposed over different gaps. Computing powers and artificial intelligence are largely behind these spaces. With a massive amount of speech recognition has hit an inflection point where its capabilities are roughly onpar with humans.

The first speech recognition systems were focused on numbers, not words in the late 1950 and 1960s. In 1952 Bell Laboratories designed the “Audrey” system which could recognize a single voice speaking digit allowed. In 2011, Apple launched ‘Siri’ which was similar to Google Voice Search, the early part of the decade saw an explosion of some other voice recognition apps and software. Today,some of the largest tech components are competing to herald the speech accuracy title.

This technology voice application is now relatively inexpensive and powerful, with advancements in AI and the increasing amounts of speech data that can be mined, it is very possible that voice becomes the next dominant interface.

1.4 Objective

1. This system helps to know about the Voice Assistant and Notepad interface is feasible and crucial forusability, efficiency, correctness search tells me about how this interface will help to propagate neural networking, machine learning, artificial intelligence, and Python modules and Python programs and further search.

2. Decreased billable hours on searching.

3. Improved productivity and a more streamlined workflow for the entire team Morework is done throughout the work week so working after hours and on weekends becomes a thing of the past.

4. No more missing important events because you have to work.

5. It provides support information for any queries in the mind of a beginner who has just started any new topic.

Chapter 2: METHODOLOGY

2.1 Introduction of Methodology

Research indicates that many traditional AI development methodologies are based on outmoded concepts dating back to the 1970s. These methodologies are being utilized to develop websites and, not surprisingly, they are limited since they were never intended to be used for this purpose. Before moving on to put forward methodology for speech recognition, it is worth considering traditional AI methodologies and their applicability to this process. When it comes to our interactions with machines, things have gotten a lot more complicated. We've gone from large mechanical buttons to touchscreens. However, hardware isn't the only thing that's changing. Throughout the history of computers, text has been the primary method of input. But thanks to developments in NLP and ML (Machine Learning), Data Science, and Artificial Intelligence we now have the means to use speech as a medium for interacting with our gadgets in the near future.

This project is an activity that helps us to improve our learning, planning and critical thinking ability. Here in Voice Assistant and Notepad is an advanced speech recognition system that utilizes cutting edge technology to provide accurate and intelligent speech recognition capabilities. Our system hosts a range of powerful set of commands which will enhance your speech recognition experience. User have to choose whether they want Voice Assistant or Voice notepad to run in the Welcome Interface. And then give commands accordingly.

2.2 Project Structure:

The system will get all the commands from the user/client asking him to perform either the VOICE NOTEPAD or the VOICE ASSISTANT in the welcome interface. Then the program will run according to the input. In Voice Notepad the user/client will have to press certain buttons on the GUI to perform certain tasks and in the Voice Assistant the user have to give certain commands to perform. Then the system will search for the query if it is defined in it. Then it will perform that certain task and bring the result. The result will be printed on the screen or will take the user to the desired website. Picture 2.1 shows how the project is running.

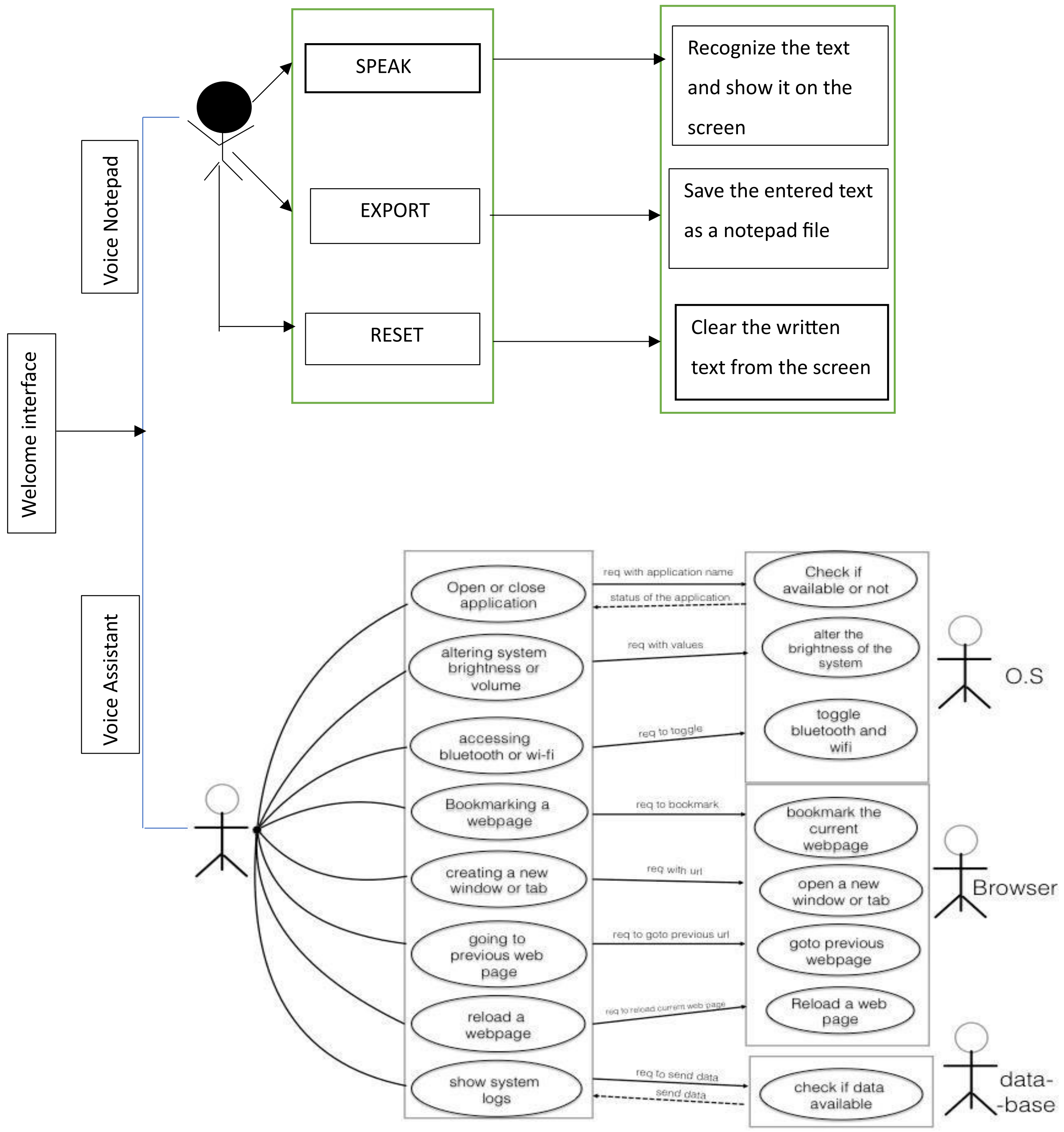

Voice Notepad
Welcome interface
Voice Assistant
SPEAK
EXPORT
RESET
Recognize the text and show it on the screen
Save the entered text as a notepad file
Clear the written text from the screen
Open or close application
altering system brightness or volume
accessing bluetooth or wi-fi
Bookmarking a webpage
creating a new window or tab
going to previous web page
reload a webpage
show system logs
req with application name
status of the application
req with values
req to toggle
req to bookmark
req with url
req to goto previous url
req to reload current web page
req to send data
send data
Check if available or not
alter the brightness of the system
toggle bluetooth and wifi
bookmark the current webpage
open a new window or tab
goto previous webpage
Reload a web page
check if data available
O.S
Browser
data-
-base

Figure 2.1: Project Structure of the Voice Assistant and Voice Notepad

2.3 Algorithm for the Voice Assistant, “BOT”:

1. Initialize the Voice Assistant system.
2. Load necessary libraries and dependencies for speech recognition and natural language processing.
3. Set up the audio input device for capturing speech.
4. Continuously listen for user input by capturing audio.
5. Preprocess the captured audio to remove noise and enhance speech quality.
6. Convert the pre-processed audio into text using a speech recognition algorithm.
7. Perform natural language processing on the recognized text to extract the user's intent and important keywords.
8. Determine the type of task or query based on the user's intent and keywords.
9. Execute the appropriate action or provide a response based on the determined task type.
10. If the task requires retrieving information, connect to relevant data sources or APIs to fetch the required data.
11. Apply any necessary algorithms or techniques to process the fetched data and generate meaningful insights or responses.
12. Generate a human-readable response or perform the requested action based on the processed data.
13. If the task is completed, go back to step 4 to listen for new user input. Otherwise, continue to step 14.
14. If there is an error or the user input is not understood, provide appropriate error handling or clarification prompts.
15. Allow for voice-based interaction to gather additional information or clarify user intent, if necessary.
16. Update the system's knowledge base or models based on user interactions and feedback to improve future responses.
17. Continue to loop through steps 4 to 16, providing responses and interacting with the user as needed.
18. Terminate the BOT system when the interaction with the user ends or when explicitly instructed to stop.

Chapter 3: IMPLEMENTATION:

3.1 About Voice Assistant and Notepad:

This Voice Assistant and Notepad is an advanced AI system designed to provide speech recognition and natural language understanding capabilities. It combines various technologies like audio preprocessing, speech recognition algorithms, and Natural Language Processing (NLP) to enable searches communication between users and the AI system. The primary goal of this project is to understand and interpret spoken language, allowing users to interact with the systems using their voices. By capturing audio input from the users, the system processes the speech through sophisticated algorithms that remove noise, enhance speech quality and convert the audio into text using speech recognition text.

To start the program the user have to run the "Welcome" python file in the command prompt (cmd). Once they run it, it will ask the user to enter the user their choices based on their need, either Voice Notepad or Voice Assistant. If the user continues with Voice Notepad, then the system will open an interface which has three buttons, options. If you click on Speak the program turn the audio into text and shows it in it's window. If you want to save the text you will have to click on the Export button and it will create a notepad file with a random name and save the text in it. And if you want to clear the text window the you will have to press the Reset button.

Now, If the user chooses the second option, Voice assistant then it will run through a command line interface. Once the audio is transformed into text, BOT employs NLP algorithms to analyze and extract the intent and important keywords from the user's speech. This enables the system to understand the user's commands, queries or requests. The extracted intent and keywords are then used to determine the type of tasks or action required. Based on the determined task, BOT can perform various actions such as retrieving information from data sources or APIs, processing fetched data using algorithms, generating meaningful insights or responses, and executing appropriate actions based on user commands. It also incorporates error-handling mechanisms and clarification prompts to address any misunderstandings or uncertainties that may rise during conversations. Additionally, the system can engage in voice-based interaction with users to gather additional information to find the best results possible. The flexibility, accuracy and adaptability of BOT make it a valuable tool for enabling a seamless speech-driven user experience.

Using the speech recognition and natural language understanding capabilities it offers a user friendly and efficient means of interacting with AI systems. It find applications in various domains, such as virtual assistants, voice-controlled systems, and more. The benefits of this system are limitless making it into a great system to interact with the AI and enhancing the experience of user.

3.2 Modules Used:

I. Tkinter:

Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it’s the only framework built into the Python standard library.

II. SpeechRecognition:

Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. You can even program some devices to respond to these spoken words.

III. pyttsx3:

It is a text to speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3.

IV. OS:

The OS module in Python provides functions for interacting with the operating system. OS comes under python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality.

V. Subprocess:

Subprocess in python is a module used to run new codes and applications by creating new processes. It let’s you start new application right from the Python program you are currently writing.

VI. Wolfram Alpha:

Wolfram Alpha is an API which can compute expert-level answers using wolfram’s a algorithms, knowledgebase and AI technology. It is made possible by the Wolfram Language.

VII. Wikiquote:

This package is for Python 3.X which allows you to search and retrieve quotes from any wikiquote article, as well as retrieve the quote of the day.

VIII. json:

JavaScript Object Notation (JSON) is a standardized format commonly used to transfer data as text that can be sent over a network.

IX. Wikipedia:

Wikipedia is a python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more.

X. datetime:

datetime in Python is the combination between dates and times. The attributes of the class are similar to both date and separate classes.

XI. webbrowser:

The webbrowser module provides a high-level interface to allow displaying web-based documents to users.

XII. winshell:

The winshell module is a light wrapper around the Windows shell functionality. It includes convenience functions for accessing special folders, for using shell's file copy, rename & delete functionality and certain amount of support for structured storage.

XIII. pyjokes:

As the name suggests, this function is used to actually return a single joke from a certain category and in a particular language.

XIV. smtplib:

The smtplib module defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP or ESMTP listener daemon.

XV. ctypes:

ctypes is a foreign function library for python. It provides C compatible datatypes, and allows calling function in DLLs or shared libraries.

XVI. requests:

The requests module allows you to send HTTP requests using Python.

XVII. getpass:

The getpass module provides a platform-independent way to enter a password in a command-line program.

XVIII. wmi:

WMI, abbreviated for Windows Management Instrumentation, is an implementation of Microsoft to the Common Information Model (short for CMI) for the DMTF, which is a vendor-neutral, industry standard method of demonstrating the Management Information.

XIX. pathlib:

Python's pathlib module provides a modern and Pythonic way of working with file paths, making code more readable and maintainable.

XX. clint:

Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. You can even program some devices to respond to these spoken words.

XXI. Selenium:

Selenium is a powerful tool for controlling web browsers through programs and performing browser automation.

XXII. ecapture:

The module is used to capture images from your camera. This module comes built-in with Python.

3.3 Language Used:

Python is a high-level, general-purpose, and very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

We are using Python to write the whole code. There are several reasons for using this language, stated below-

• **Simplicity and Readability:** Python has a clean and easy-to-read syntax, making it beginner-friendly and highly readable. This makes it easier for developers to understand and maintain the codebase.

• **Extensive Libraries and Frameworks:** Python offers a wide range of libraries and frameworks specifically designed for AI and machine learning tasks. Libraries like TensorFlow, PyTorch, and scikit-learn provide powerful tools for building and training AI models.

• **Vibrant Ecosystem:** Python has a vibrant and active community that actively develops and maintains various AI-related libraries, tools, and resources. This means that developers have access to a wealth of resources, documentation, and community support.

• **Interoperability:** Python can easily integrate with other programming languages and technologies. This flexibility allows developers to leverage existing systems, libraries, and APIs, making it easier to connect SRISHTI with various data sources or APIs for fetching information.

• **Data Processing Capabilities:** Python offers robust data processing and manipulation capabilities through libraries like NumPy and pandas. These libraries provide efficient data structures and operations, which are crucial for tasks like preprocessing audio, and text, and working with structured data.

• **Prototyping and Rapid Development:** Python's simplicity and extensive libraries make it an ideal choice for prototyping and rapid development of AI systems like SRISHTI. It allows developers to quickly test and iterate on ideas, making the development process more efficient.

• **Community and Industry Support:** Python is widely adopted in both the AI research community and the industry. Many AI-related conferences, workshops, and competitions use Python as the primary language. This support and recognition make Python a natural choice for AI projects.

3.4 Resources used for this project:

3.4.1 Software Used-

- Operating System: Windows 10 Pro
- Python Pycharm Community Edition

3.4.2 Hardware Used-

- CPU: AMD RYZEN 5 1600X
- Ram: 8GB
- Hard Disk Memory:1 TB

Chapter 4: ANNEXURE (Source Code)

Welcome.py:

```
import os

import pyttsx3

from sys import platform


engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

engine.setProperty('voices', voices[1].id)


def speak(audio):

    engine.say(audio)

    engine.runAndWait()


def banner():

    speak("Welcome to our project!")

    print("1. Voice Notepad\n2. Voice Assistant")

    speak("Please enter your choice:")

    s = input("Please enter your choice: ")


if banner() == "1" or banner() == "Voice Notepad" or banner() == "Notepad" or banner()
== "notepad":

    if platform == "linux" or platform == "linux2":

        os.system("Linux/voicenote.py")
    elif platform == "win32":
        os.system("Windows/voicenote.py")

elif banner() == "Voice Assistant" or banner() == "2" or banner() == "Assistant" or
banner() == "assistant":
    os.system("PythonProjects.py")
else:
    print()
```

VoiceNotepad.py:

```
# Import tkinter for User Interface creation
# Import speech_recognition for google speech recognition services
# Import time for current date time
# Import os for get current working directory

import tkinter as tk
from tkinter import *
import speech_recognition as sr
import time
from time import ctime
import os

# Create root windows
root = tk.Tk()
```

```
# Set Title of the Application

root.title("Voice Notepad")

# Set Application Icon

root.iconbitmap(r'speech.ico')

# Fix size of the windows

root.resizable(width=False, height=False)


## Create Frame for Buttons

# First parameter-> main windows(or root), bg-> Frame background

frame_button = Frame(root, bg = 'blue')

frame_button.pack(side = LEFT, fill=BOTH)

# Create Frame for TextArea

frame_textarea = Frame(root, bg = 'yellow')

frame_textarea.pack(side = LEFT, fill=BOTH)


# Create Textarea with given hight, width and padding within frame frame_textarea

# First parameter-> in which frame, height-> height of the textarea, width-> width of the
textarea

TextArea = Text(frame_textarea, height=21, width=50)

TextArea.pack(padx=20, pady=20)


# Create Images for Set on Button

image_mike = tk.PhotoImage(file="mike.png")

image_export = tk.PhotoImage(file="download.png")
```

```
image_reset = tk.PhotoImage(file="reset.png")


# Method to clear Textarea
def clearScreen():
    TextArea.delete('1.0', END)


# Create method to google speech recognition
def convertSpeechToText():
    # call google speech recognition
    r = sr.Recognizer()
    with sr.Microphone() as source:
        audio = r.listen(source)


    # Speech recognition using Google Speech Recognition
    data = ""
    insert_data = ""
    try:
        # Uses the default API key
        # To use another API key: `r.recognize_google(audio,
key="GOOGLE_SPEECH_RECOGNITION_API_KEY")`
        data = r.recognize_google(audio)


        insert_data = insert_data + " " + data
        # Write speech to text converted data into Textarea
        TextArea.insert(INSERT, insert_data)
    except sr.UnknownValueError:
        # Catch and print exception for speech which was not recognised by google API
        insert_data="Google Speech Recognition could not understand audio"
        TextArea.insert(INSERT, insert_data)
    except sr.RequestError as e:
        # Catch and print exception for failed to call google API
        insert_data="Could not request results from Google Speech Recognition service;
{0}".format(e)
        TextArea.insert(INSERT, insert_data)



# Method to write Textarea data into a new text file
def writeToFile():
    # get speech to text converted data from Textarea
    speech_data=TextArea.get(1.0,END)[:-1]
    # speech_data= speech_data.strip()

    # get current directory in which program stored
    save_path = os.getcwd()
    # get current time stampt to use for file name
    name_of_file = getFileName()
    # file name with directory name and extention
    completeName = os.path.join(save_path, name_of_file+".txt")
    # Open file
    file1 = open(completeName, "w")
    # write Textarea data into the file
    file1.write(speech_data)
    # Close file
    file1.close()

# Method to get current timestamp
def getFileName():
    ts = time.time()
    # Convert floating point timestamp into integer
    ts=int(ts)
    # Convert Integer into String
    ts=str(ts)
    # return current timestamp string
    return ts



# Create button for Export Textarea data into text file, Enable google speech to text API
and call Clear Text area
# First parameter-> in which Frame, text->text show on button, image-> for set Image on
Button, compound->top or left(placement of image over text on button)
# command->method which called on button press, height-> height of the button, width->
width of the button
```

```
btn_export = Button(frame_button, text='Export', image=image_export,
compound="top", command=writeToFile, height=80, width=80)

btn_export.pack(pady = 20, padx = 20)

btn_speak = Button(frame_button, text="Speak", image=image_mike, compound="top",
command=convertSpeechToText, height=80, width=80)

btn_speak.pack(pady = 20, padx = 20)

btn_reset = Button(frame_button, text="Reset", image=image_reset, compound="top",
command=clearScreen, height=80, width=80)

btn_reset.pack(pady = 20, padx = 20)


# The method mainloop has an important role for TkInter, it is waiting for events and
updating the GUI

root.mainloop()
```

VoiceAssistant.py:

```
import subprocess

import wolframalpha

import wikiquote

import pyttsx3

import json

import speech_recognition as sr

import datetime

import wikipedia

import webbrowser

import os
import pyautogui

import keyboard

import winshell

import pyjokes

import feedparser

import smtplib

import ctypes

import time

import requests

import fileinput

import getpass

import wmi

import os

from pathlib import Path

from clint.textui import progress

from selenium import webdriver

from ecapture import ecapture as ec

from bs4 import BeautifulSoup

import win32com.client as wincl

from urllib.request import urlopen

from pytube import YouTube



engine = pyttsx3.init("sapi5")
```

```
voices = engine.getProperty("voices")
engine.setProperty("voice", voices[0].id)


DIRECTORIES = {
    "HTML": [".html5", ".html", ".htm", ".xhtml"],
    "IMAGES": [
        ".jpeg",
        ".jpg",
        ".tiff",
        ".gif",
        ".bmp",
        ".png",
        ".bpg",
        "svg",
        ".heif",
        ".psd",
    ],
    "VIDEOS": [
        ".avi",
        ".flv",
        ".wmv",
        ".mov",
        ".mp4",
        ".webm",
        ".vob",
        ".mng",
        ".qt",
        ".mpg",
        ".mpeg",
        ".3gp",
        ".mkv",
    ],
    "DOCUMENTS": [
        ".oxps",
        ".epub",
        ".pages",
        ".docx",
        ".doc",
        ".fdf",
        ".ods",
        ".odt",
        ".pwi",
        ".xsn",
        ".xps",
        ".dotx",
        ".docm",
        ".dox",
        ".rvg",
        ".rtf",
        ".rtfd",
        ".wpd",
        ".xls",
        ".xlsx",
        ".ppt",
        "pptx",
    ],
    "ARCHIVES": [
        ".a",
        ".ar",
        ".cpio",
        ".iso",
        ".tar",
        ".gz",
        ".rz",
        ".7z",
        ".dmg",
        ".rar",
        ".xar",
        ".zip",
    ],
    "AUDIO": [
        ".aac",
        ".aa",
        ".aac",
        ".dvf",
        ".m4a",
        ".m4b",
        ".m4p",
        ".mp3",
        ".msv",
        "ogg",
        "oga",
        ".raw",
        ".vox",
        ".wav",
        ".wma",
    ],
    "PLAINTEXT": [".txt", ".in", ".out"],
    "PDF": [".pdf"],
    "PYTHON": [".py", ".pyi"],
    "XML": [".xml"],
    "EXE": [".exe"],
    "SHELL": [".sh"],
}
FILE_FORMATS = {
    file_format: directory
    for directory, file_formats in DIRECTORIES.items()
    for file_format in file_formats
}


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def countdown(n):
    while n > 0:
        print(n)
        n = n - 1
    if n == 0:
        print("BLAST OFF!")


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

    elif hour >= 12 and hour < 18:
        speak("Good Afternoon Sir!")

    else:
        speak("Good Evening Sir!")

    assname = "bot"
    speak("I am your Assistant")
    speak(assname)


def usrname():
    speak("What should i call you sir")
    uname = takeCommandname()
    speak("Welcome Mister")
    speak(uname)
    print("#####################")
    print("Welcome Mr.", uname)
    print("#####################")


def quotaton():
    speak(wikiquote.quote_of_the_day())
    print(wikiquote.quote_of_the_day())
```

```
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language="en-in")
        print(f"User said: {query}\n")

    except Exception as e:
        print(e)
        print("Unable to Recognizing your voice.")
        return "None"
    return query



def takeCommandname():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Username...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Trying to Recognizing Name...")
        query = r.recognize_google(audio, language="en-in")
        print(f"User said: {query}\n")

    except Exception as e:
        print(e)
        print("Unable to Recognizing your name.")
        takeCommandname()
        return "None"
    return query



def takeCommandmessage():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Enter Your Message")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        query = r.recognize_google(audio, language="en-in")
        print(f"Message to be sent is : {query}\n")

    except Exception as e:
        print(e)
        print("Unable to recognize your message")
        print("Check your Internet Connectivity")
    return query



def takeCommanduser():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Name of User or Group")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        query = r.recognize_google(audio, language="en-in")
        print(f"Client to whom message is to be sent is : {query}\n")

    except Exception as e:
        print(e)
        print("Unable to recognize Client name")
        speak("Unable to recognize Client Name")
        print("Check your Internet Connectivity")
    return query



def openapps():
    speak("Ok Sir , Wait A Second!")

    if 'code' in query:
        os.startfile("C:\\Users\\Root\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe")

    elif 'Whatsapp' in query:
        os.startfile("C:\\Users\\Root\\Desktop\\WhatsApp.lnk")

    elif 'Word' in query:
        os.startfile("C:\\ProgramData\\Microsoft\\Windows\\Start
Menu\\Programs\\Word.lnk")

    elif 'chrome' in query:
        os.startfile("C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe")

    elif 'facebook' in query:
        webbrowser.open('https://www.facebook.com/')
    elif 'instagram' in query:
        webbrowser.open('https://www.instagram.com/')

    elif 'maps' in query:
        webbrowser.open('https://www.google.com/maps/@28.7091225,77.2749958,15z')

    elif 'youtube' in query:
        webbrowser.open('https://www.youtube.com')

    speak("Your Command Has Been Completed Sir!")


def closeapps():
    speak("Ok Sir , Wait A second!")

    if 'youtube' in query:
            os.system("TASKKILL /F /im Chrome.exe")

    elif 'chrome' in query:
            os.system("TASKKILL /f /im Chrome.exe")

    elif 'telegram' in query:
            os.system("TASKKILL /F /im Telegram.exe")
    elif 'code' in query:
            os.system("TASKKILL /F /im code.exe")

    elif 'instagram' in query:
            os.system("TASKKILL /F /im chrome.exe")

    speak("Your Command Has Been Succesfully Completed!")



def YoutubeAuto():
    speak("Whats Your Command ?")
    comm = takeCommand()

    if 'pause' in comm:
        keyboard.press('space bar')
    elif 'play  ' in comm:
        keyboard.press('space bar')
    elif 'restart' in comm:
        keyboard.press('0')

    elif 'mute' in comm:
        keyboard.press('m')

    elif 'skip' in comm:
        keyboard.press('l')

    elif 'back' in comm:
        keyboard.press('j')

    elif 'full screen' in comm:
        keyboard.press('f')

    elif 'film mode' in comm:
        keyboard.press('t')

    speak("Done Sir")



def screenshot():
    speak("Ok Boss , What Should I Name That File ?")
    path = takeCommand()
    path1name = path + ".png"
    path1 = "C:\\Users\\user\\Desktop\\Speech_To_Text\\Voice-Notepad\\Windows" +
path1name
    kk = pyautogui.screenshot()
    kk.save(path1name)
    os.startfile("C:\\Users\\user\\Desktop\\Speech_To_Text\\Voice-Notepad\\Windows")
    speak("Here Is Your ScreenShot")
```

```
def takeCommandcontent():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("What Should i say, sir")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        query = r.recognize_google(audio, language="en-in")
        print(f"Message to be sent is: {query}\n")

    except Exception as e:
        print(e)
        print("Unable to recognize")
    return query



def organize():
    for entry in os.scandir():
        if entry.is_dir():
            continue
        file_path = Path(entry.name)
        file_format = file_path.suffix.lower()
        if file_format in FILE_FORMATS:
            directory_path = Path(FILE_FORMATS[file_format])
            directory_path.mkdir(exist_ok=True)
            file_path.rename(directory_path.joinpath(file_path))
    try:
        os.mkdir("OTHER")
    except:
        pass
    for dir in os.scandir():
        try:
            if dir.is_dir():
                os.rmdir(dir)
            else:
                os.rename(
                    os.getcwd() + "/" + str(Path(dir)),
                    os.getcwd() + "/OTHER/" + str(Path(dir)),
                )
        except:
            pass


def sendEmail(to, content):
    server = smtplib.SMTP("smtp.gmail.com", 587)
    server.ehlo()

    server.starttls()

    server.login("debargha51@gmail.com", "debarghanandi")

    server.sendmail("debarghashanewatson2001@gmail.com", to, content)

    server.close()



def youtubeauto():

    speak("Whats Your Command ?")

    comm = takeCommand()


    if 'pause' in comm:

        keyboard.press('space bar')

    elif 'play  ' in comm:

        keyboard.press('space bar')

    elif 'restart' in comm:

        keyboard.press('0')


    elif 'mute' in comm:

        keyboard.press('m')


    elif 'skip' in comm:

        keyboard.press('l')
    elif 'back' in comm:
        keyboard.press('j')

    elif 'full screen' in comm:
        keyboard.press('f')

    elif 'film mode' in comm:
        keyboard.press('t')

    speak("Done Sir")


def chromeauto():
    speak("Chrome Automation started!")
    command = takeCommand()

    if 'close this tab' in command:
        keyboard.press_and_release('ctrl + w')

    elif 'open new tab' in command:
        keyboard.press_and_release('ctrl + t')

    elif 'open new window' in command:
        keyboard.press_and_release('ctrl + n')
    elif 'history' in command:
        keyboard.press_and_release('ctrl +h')



if __name__ == "__main__":
    clear = lambda: os.system("cls")
    clear()
    wishMe()
    usrname()
    speak("Can i tell you a quote of day")
    useropt = takeCommand().lower()
    if "yes" in useropt or "sure" in useropt:
        quotaton()
    else:
        speak("Taking you to command function")

    speak("How can i Help you, Sir")
    while True:
        query = takeCommand().lower()
        assname = "bot"
        if "wikipedia" in query:
            speak("Searching Wikipedia...")
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=3)
            speak("Answer From Wikipedia")
            print(results)
            speak(results)

        elif "Good Morning" in query:
            speak("A warm" + query)
            speak("How are you Mister")
            speak(assname)

        elif "wikipedia" in query and "hindi" in query:
            speak("Searching Wikipedia...")
            query = query.replace("wikipedia", "")
            query = query.replace("hindi", "")
            results = wikipedia.summary(query, sentences=3)
            speak("According to Wikipedia")
            r = sr.Recognizer()
            results = r.recognize_google(results, language="hi")
            print(results)
            speak(results)

        elif "open youtube" in query:
            speak("Taking You To Youtube\n")
            webbrowser.open("https://youtube.com")
```

```
elif "open google" in query:
    speak("Taking you to Google\n")
    webbrowser.open("https://google.com")

elif "change brightness to " in query:
    query = query.replace("change brightness to", "")
    brightness = query
    c = wmi.WMI(namespace="wmi")
    methods = c.WmiMonitorBrightnessMethods()[0]
    methods.WmiSetBrightness(brightness, 0)

elif "Organize Files" in query:
    organize()

elif "open stackoverflow" in query:
    speak("Here you go to Stack Over flow.Happy coding")
    webbrowser.open("https://stackoverflow.com/")

elif "stackoverflow " in query:
    speak("Stackoverflow khola ja rha h")
    webbrowser.open("https://stackoverflow.com/")

elif "send a whatsapp message" in query or "send a WhatsApp message" in query:
    driver = webdriver.Chrome("Web Driver Location")
    driver.get("https://web.whatsapp.com/")
    speak("Scan QR code before proceding")
    tim = 10
    time.sleep(tim)
    speak("Enter Name of Group or User")
    name = takeCommanduser()
    speak("Enter Your Message")
    msg = takeCommandmessage()
    count = 1
    user = driver.find_element_by_xpath('//span[@title = "{}"]'.format(name))
    user.click()
    msg_box = driver.find_element_by_class_name("_3u328")
    for i in range(count):
        msg_box.send_keys(msg)
        button = driver.find_element_by_class_name("_3M-N-")
        button.click()

elif "stackoverflow " in query:
    speak("Stackoverflow khola ja rha h")
    webbrowser.open("https://stackoverflow.com")

elif (
    "play music" in query
    or "play song" in query
    or "gaana" in query
    or "song" in query
  ):
    # music_dir = "G:\\Song"
    username = getpass.getuser()
    music_dir = "C:\\Users\\" + username + "\\Music"
    songs = os.listdir(music_dir)
    print(songs)
    random = os.startfile(os.path.join(music_dir, songs[1]))

  elif "the time" in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"Sir, the time is {strTime}")

  elif "samay" in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"samaye hai {strTime}")

  # elif 'open opera' in query:
  # codePath =
r"C:\\Users\\GAURAV\\AppData\\Local\\Programs\\Opera\\launcher.exe"
  # os.startfile(codePath)
```

```
elif "email to deb" in query:
    try:
        content = takeCommandcontent()
        to = "debarghashanewatson2001@gmail.com"
        sendEmail(to, content)
        speak("Email has been sent!")
    except Exception as e:
        print(e)
        speak("I am not able to send this email")

elif "how are you" in query:
    speak("I am fine , Thank you")
    # speak("How are you, Sir")

# elif "change my name to" in query:
# query=query.replace("change my name to","")
# assname=query

elif "change name" in query:
    speak("What would you like to call me ,Sir ")
    assname = takeCommand()
    speak("Thanks for naming me")

elif "what's your name" in query or "What is your name" in query:
    speak("My friends call me")

    speak(assname)

    print("My friends call me", assname)


elif "exit" in query:

    speak("Thanks for giving me your time")

    exit()


elif "who made you" in query or "who created you" in query:

    speak("I have been created by Debargha,Manju and Pinaki.")


elif "joke" in query:

    speak(pyjokes.get_joke())


elif "calculate" in query:

    app_id = "KRT8KQ-TTH3Y4LRYH"

    client = wolframalpha.Client(app_id)

    indx = query.lower().split().index("calculate")

    query = query.split()[indx + 1 :]

    res = client.query(" ".join(query))

    answer = next(res.results).text

    print("The answer is " + answer)

    speak("The answer is " + answer)
```

```
        elif "search" in query or "play" in query:
            query = query.replace("search", "")
            query = query.replace("play", "")
            webbrowser.open(query)

        elif "who am i" in query:
            speak("If you talk then definately your human.")

        elif "why you came to world" in query:
            speak("Thanks to Debargha . To gain control over Humans")

        elif "What is love" in query:
            speak("It is 7th sense that destroy all other senses")

        elif "who are you" in query:
            speak("I am your virtual assistant created by Debargha")

        elif "reason for you" in query:
            speak("I was created as a Minor project by Mister Debargha and his friends ")

        elif "change background" in query:
            ctypes.windll.user32.SystemParametersInfoW(
                20, 0, "C:/Users/user/Desktop/project/Python-Voice-
Assistant/background.jpeg", 0
```

```
    )
    speak("Background changed succesfully")

elif "open bluestack" in query:
    appli = r"C:\Users\Public\Desktop\BlueStacks 5"
    os.startfile(appli)

elif 'screenshot' in query:
    screenshot()

elif 'open facebook' in query:
    openapps()

elif 'open instagram' in query:
    openapps()

elif 'open maps' in query:
    openapps()

elif 'open code' in query:
    openapps()

elif 'open youtube' in query:
    openapps()
```

```
elif 'open WhatsApp' in query:

    openapps()


elif 'open Word' in query:

    openapps()


elif 'open chrome' in query:

    openapps()


elif 'close chrome' in query:

    closeapps()


elif 'close telegram' in query:

    closeapps()


elif 'close instagram' in query:

    closeapps()


elif 'close facebook' in query:

    closeapps()


elif 'pause' in query:

    keyboard.press('space bar')
```

```
elif 'restart' in query:
    keyboard.press('0')

elif 'mute' in query:
    keyboard.press('m')

elif 'skip' in query:
    keyboard.press('l')

elif 'back' in query:
    keyboard.press('j')

elif 'full screen' in query:
    keyboard.press('f')

elif 'film mode' in query:
    keyboard.press('t')

elif 'youtube tool' in query:
    youtubeauto()

elif 'close the tab' in query:
    chromeauto()
```

```
elif 'open new tab' in query:
    chromeauto()

elif 'open new window' in query:
    chromeauto()

elif 'history' in query:
    chromeauto()

elif 'close youtube' in query:
    os.system("TASKKILL /F /im Chrome.exe")

elif 'close chrome' in query:
    os.system("TASKKILL /f /im Chrome.exe")

elif 'close telegram' in query:
    os.system("TASKKILL /F /im Telegram.exe")

elif 'close code' in query:
    os.system("TASKKILL /F /im code.exe")

elif 'close instagram' in query:
    os.system("TASKKILL /F /im chrome.exe")
```

```
elif "google news" in query:
    try:
        jsonObj = urlopen(
            "https://newsapi.org/v2/top-headlines?sources=google-news-in&"
            "apiKey=9f3720c8d51f425ba93f8abc78e9ea29"
        )
        data = json.load(jsonObj)
        i = 1
        speak("")
        print("""===============Google News============""" + "\n")
        for item in data["articles"]:
            print(str(i) + ". " + item["title"] + "\n")
            print(item["description"] + "\n")
            speak(str(i) + ". " + item["title"] + "\n")
            i += 1
    except Exception as e:
        print(str(e))

elif "bbc news" in query:
    try:
        main_url = " https://newsapi.org/v1/articles?source=bbc-news&sortBy=top&" \
                   "apiKey=9f3720c8d51f425ba93f8abc78e9ea29"
        open_bbc_page = requests.get(main_url).json()
        article = open_bbc_page["articles"]
        results = []
        for ar in article:
            results.append(ar["title"])
        for i in range(len(results)):
            print(i + 1, results[i])
    except Exception as e:
        print(str(e))

elif "news" in query:  # samachar
    try:
        jsonObj = urlopen(
            "https://newsapi.org/v1/articles?source=the-times-of-india&sortBy=top&"
            "apiKey=9f3720c8d51f425ba93f8abc78e9ea29"
        )
        data = json.load(jsonObj)
        i = 1
        speak("here are some top news from the times of india")
        print("""===============TIMES OF INDIA============""" + "\n")
        for item in data["articles"]:
            print(str(i) + ". " + item["title"] + "\n")
            print(item["description"] + "\n")
            speak(str(i) + ". " + item["title"] + "\n")
            i += 1
    except Exception as e:
        print(str(e))

elif "lock window" in query or "system ko lock Karen" in query:
    speak("locking the device")
    ctypes.windll.user32.LockWorkStation()

elif "shutdown system" in query:
    speak("Hold On a Sec! Your system is on its way to shut down")
    subprocess.call("shutdown /p /f")

elif "empty recycle bin" in query:
    winshell.recycle_bin().empty(confirm=False, show_progress=False, sound=True)
    speak("Recycle Bin Recycled")

elif "don't listen" in query or "stop listening" in query:
    speak("for how much time you want to stop jarvis from listening commands")
    a = int(takeCommand())
    time.sleep(a)
    print(a)

elif "where is" in query:
    query = query.replace("where is", "")
    location = query
    speak("Locating ")
    speak(location)
    webbrowser.open("https://www.google.nl/maps/place/" + location + "")

elif "camera" in query or "take a photo" in query:
    ec.capture(0, "Bot Camera ", "img.jpg")

elif "restart" in query:
    subprocess.call(["shutdown", "/r"])

elif "hibernate" in query or "sleep" in query:
    speak("Hibernating")
    subprocess.call("shutdown /i /h")

elif "log off" in query or "sign out" in query:
    speak("Make sure all the application are closed before sign-out")
    time.sleep(5)
    subprocess.call(["shutdown", "/l"])

elif "countdown of" in query:
    query = int(query.replace("countdown of ", ""))
    countdown(query)

elif "write a note" in query:
    speak("What should i write , sir")
    note = takeCommand()
    file = open("Bot.txt", "w")
    speak("Sir, Should i include date and time")
    snfm = takeCommand()
    if "yes" in snfm or "sure" in snfm:
        strTime = datetime.datetime.now().strftime("%H:%M:%S")
        file.write(strTime)
        file.write(" :- ")
        file.write(note)
    else:
        file.write(note)

elif "show note" in query:
    speak("Showing Notes")
    file = open("Bot.txt", "r")
    print(file.read())
    speak(file.read(6))

elif "update assistant" in query:
    speak(
        "After downloading file please replace this file with the downloaded one"
    )
    url = "#url after uploading file"
```

```
        r = requests.get(url, stream=True)
        with open("Voice.py", "wb") as Pypdf:
            total_length = int(r.headers.get("content-length"))
            for ch in progress.bar(
                r.iter_content(chunk_size=2391975),
                expected_size=(total_length / 1024) + 1,
            ):
                if ch:
                    Pypdf.write(ch)

    elif "bot" in query:
        wishMe()
        speak("Bot at your service")
        speak(assname)

    elif "weather" in query:
        api_key = "605518f0365eabcf6b6f68df0ddbe3c8"

        base_url = "http://api.openweathermap.org/data/2.5/weather?"
        speak(" City name ")
        print("City name : ")
        city_name = takeCommand()
        complete_url = base_url + "appid=" + api_key + "&q=" + city_name
        response = requests.get(complete_url)
```

```
   x = response.json()
   if x["cod"] != "404":
      y = x["main"]
      current_temperature = y["temp"]
      current_pressure = y["pressure"]
      current_humidiy = y["humidity"]
      z = x["weather"]
      weather_description = z[0]["description"]
      print(
         " Temperature (in kelvin unit) = "
         + str(current_temperature)
         + "\n atmospheric pressure (in hPa unit) ="
         + str(current_pressure)
         + "\n humidity (in percentage) = "
         + str(current_humidiy)
         + "\n description = "
         + str(weather_description)
      )
   else:
      speak(" City Not Found ")

elif "wikipedia" in query:
   webbrowser.open("wikipedia.com")
```

```
elif (
    "will you be my gf" in query or "will you be my bf" in query
):  # most asked question from google Assistant
    speak("I'm not sure about , may be you should give me some time")

elif "how are you" in query:
    speak("I'm fine, glad you asked me that")

elif "i love you" in query:
    speak("It's hard to understand")

elif "what is" in query or "who is" in query:
    client = wolframalpha.Client("KRT8KQ-U387QJ236A")
    res = client.query(query)
    try:
        print(next(res.results).text)
        speak(next(res.results).text)
    except StopIteration:
        print("No results")

elif "open Gmail" in query:
    webbrowser.open("https://mail.google.com/mail/u/0/#inbox")

elif "open yahoo mail" in query:
```

```
            webbrowser.open("https://in.mail.yahoo.com")


        elif "Show project Report" in query:
            speak("Opening Minor Project Report")
            projectre = r"C:\\Users\\user\\Desktop\\Speech_To_Text\\Voice-Notepad\\" \
                        r"Voice Assistant and Notepad report.pdf"
            os.startfile(projectre)
```

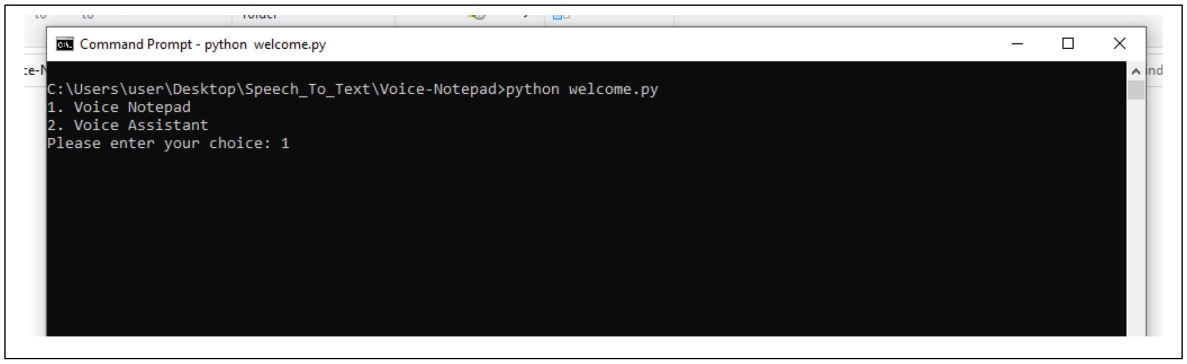
To get the access of the Full code and other additional files got to our Github account:

https://github.com/Debargha21101/Voice-Assistant-and-Notepad-

Chapter 5: RESULT AND DISCUSSION

5.1 Result and Discussion:

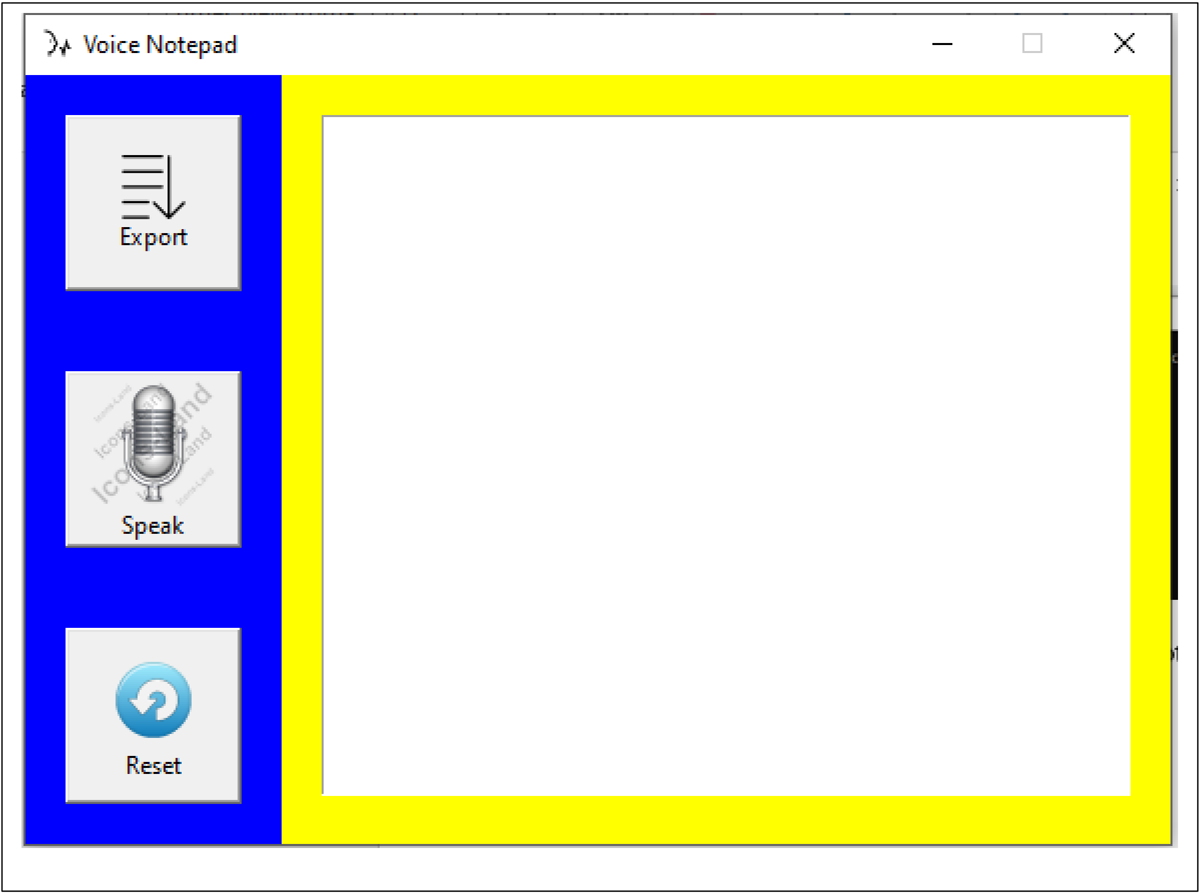
Figure 5.1.1 describes the command line interface after running the Welcome.py file from command prompt.



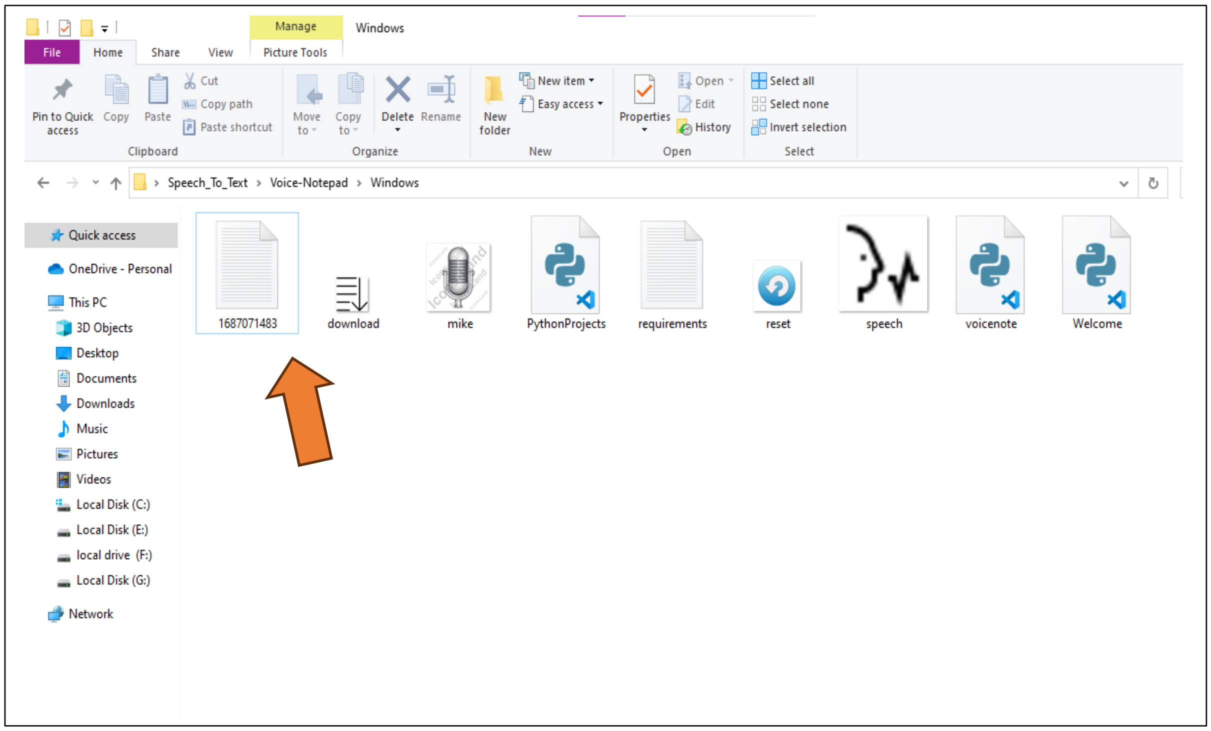
Command Prompt - python welcome.py

```
C:\Users\user\Desktop\Speech_To_Text\Voice-Notepad>python welcome.py
1. Voice Notepad
2. Voice Assistant
Please enter your choice: 1
```

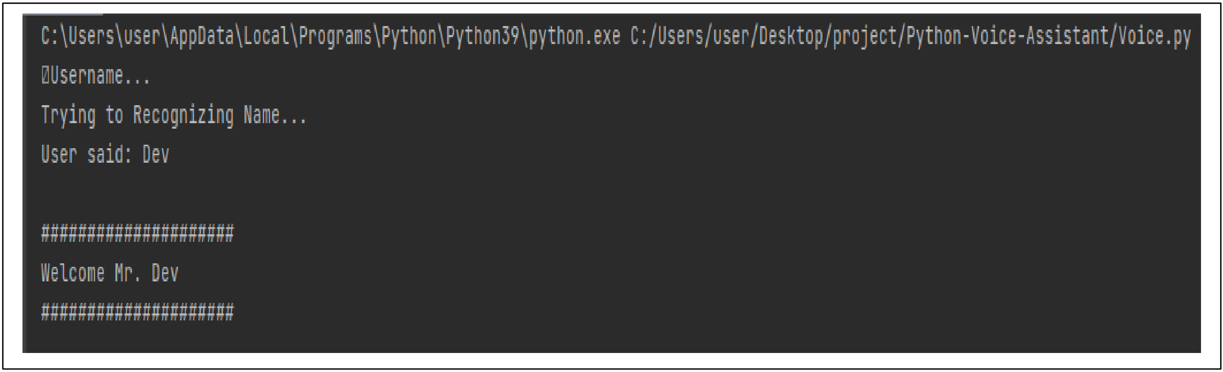
Then after giving choice to "1" or "Notepad" or "voice notepad" it will open the Voice Notepad GUI. Figure 5.1.2 will show it.



Just click on “speak” and say whatever you want. Then if you want to export the written text and save it in a notepad file click on “Export”. Figure 5.1.3 will show where the file will be created.



Now if you go back to the Welcome.py and pick option “2” or “assistant” or “Voice Assistant” it will open the command line interface of the Voice Assistant. Figure 5.1.4 will show the command line interface.



```
C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe C:/Users/user/Desktop/project/Python-Voice-Assistant/Voice.py
Username...
Trying to Recognizing Name...
User said: Dev

####################
Welcome Mr. Dev
####################
```

5.2 Limitations of this Project

- You will need a better quality of mic and speaker to enjoy this project.
- The pronunciation must be good of the user or else the project may not run properly.
- This project requires a good internet connection.

5.3 Future Work

Voice Assistant and Notepad has some great speech recognition capabilities That can greatly enhance the way we interact with technology. The system demonstrates a high level of accuracy in understanding and interpreting human language, helping users to give commands. These features help where the manual input is not possible by the user. In terms of performance, the Voice Assistant and Notepad gives satisfactory results from both ends. But like other AI systems available in the market, it may have some difficulties in understanding someone's accent, dialects or speech pattern. Continuous research is needed to improve those system performance over time. But the main aspect is that it could be used to implement with other technologies or platforms. Endless integration with voice-enabled devices, mobile applications, or customer service systems would enhance the overall user experience and extend the system's capabilities. Considering its strength and potential areas for improvement, Voice Assistant and Notepad system makes it a valuable tool for business and developers alike. With ongoing development, the Voice Assistant and Notepad has the potential to become more accurate, contributing to the advancement of speech-enabled technologies.

In the future, Voice assistant and Notepad, with the integration of Adriano, holds immense potential to redefine the way we interact with technology. Adriano, an advanced conversational Ai, brings a human like touch to this system, enhancing its capabilities and making it even more versatile and intuitive.

With Adriano's integration, Voice Assistant and Notepad will be able to engage in natural and lifelike conversations, effectively understanding and responding to user queries and requests. Adriano's advanced natural language processing abilities will enable the system to grasp complex nuances, context, and emotions, leading to more meaningful interactions. The addition of Adriano to Voice Assistant and Notepad will significantly benefit various industries and applications. In customer service, it can offer personalized and empathetic support, resolving issues and inquiries with a human-like touch. Ineducation, it can act as an interactive tutor, adapting to individual learning styles andproviding tailored explanations and guidance.

Furthermore, the integration of Adriano will open doors for improved speech recognition and synthesis capabilities. With enhanced speech recognition, the systemwill be more accurate in transcribing and understanding spoken language. Additionally, it will be capable of generating speech that closely resembles human voices, making interactions with the system more natural and immersive. As technology continues to evolve, it is crucial to ensure responsible development and address ethical considerations. This includes maintaining transparency, respecting user privacy, and ensuring proper handling of sensitive information.

In conclusion, with the integration of Adriano, the future of Voice Assistant and Notepad looks promising. The combined power of advanced conversational AI and robust speech recognition will revolutionize the way we interact with intelligent systems. Byoffering lifelike conversations and human-like responses, Voice Assistant and Notepad with Adriano hasthe potential to reshape customer service, education, and various other sectors, bringing us closer to a seamless integration of AI into our daily lives.

Chapter 6: CONCLUSION

Voice Assistant and Notepad represents an impressive technological advancement in the field of artificial intelligence. Its primary focus is on speech recognition and internal system hosting, making it a powerful tool for various applications. Voice Assistant and Notepad's ability to accurately understand and process spoken language opens up possibilities for seamless communication and interaction with machines. Whether it's voice commands, dictation, or transcription, Voice Assistant and notepad aims to provide efficient and reliable speech recognition capabilities. Additionally, Voice Assistant and Notepad's internal system hosting features enable it to serve as a robust platform for hosting and managing various AI applications and services. It provides the necessary infrastructures and resources to support the deployment and operation of intelligent systems. However, it's important to note that Voice Assistant and Notepad's effectiveness and performance depend on several factors, including data quality, training, and continuous improvement. Ongoing research anddevelopment are crucial to refine and enhance its speech recognition accuracy and system hosting capabilities.

In conclusion the Voice Assistant and Notepad serves a great potential for revolutionizing speech recognition and internal system hosting. With its advanced technology, it offers opportunities for more natural and intuitive human-machine interaction. However, further advancements and refinements are necessary to ensure optimal performance and address any limitations that may arise.

References:

1. Openai: https://openai.com/blog/chatgpt
2. Render: https://render.com/
3. GitHub: https://github.com/
4. Google: https://www.google.com/
5. News API: https://newsapi.org/
6. Weather API: https://www.weatherapi.com/
7. Cloudinary: https://cloudinary.com/

Books:

- Pattern Recognition and Machine Learning, Christopher M. Bishop.
- Dive Into Python
- Artificial Intelligence with Python: A Comprehensive guide to Building Intelligent apps for Python Beginners and Developers