

DESKTOP VOICE ASSISTANT

A Course Based Project Report Submitted in partial fulfillment of
the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

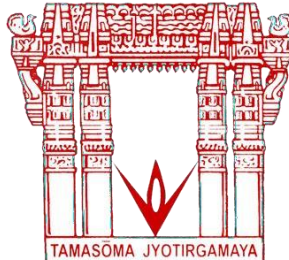
LINGALA BENJAMIN PAUL 21071A6232

KRISHNA NANDINI 21071A6231

Under the guidance of

Mrs.E.LALITHA

(Assistant Professor, Department of CSE, VNR VJIET)



DEPARTMENT OF CSE-CYS,DS and AI&DS

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE
OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited
for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE,
New Delhi,

Affiliated to JNTUH Recognized as "College with Potential for Excellence"
by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

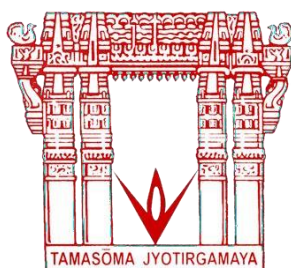
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF
ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for
CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi,
Affiliated to JNTUH Recognized as "College with Potential for Excellence" by
UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE-CYS,DS,AI&DS



CERTIFICATE

This is to certify that the project report entitled "**DESKTOP VOICE ASSISTANT**" is
a bonafide work done under our supervision and is being submitted by
L.BENJAMIN PAUL :21071A6232, KRISHNA NANDINI :21071A6231 in partial
fulfillment for the award of the degree of Bachelor of Technology in CSE-CYBER
SECURITY , of the VNRVJIET, Hyderabad during the academic year 2022-2023..

Project Guide

Mrs.E.Lalitha
Assistant prof. and internal guide
Dept. of CSE-CYS,DS,AI&DS
VNR VJIET

Head of Department

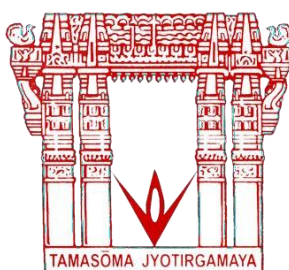
Dr. Mr.Rajasekhar
Prof. and Head
Dept. of CSE-CYS, DS and AI&DS
VNRVJIET

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE,
EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi,
Affiliated to JNTUH Recognized as "College with Potential for Excellence" by
UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE



DECLARATION

This is to certify that the project work entitled "DESKTOP VOICE ASSISTANT" submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfilment of requirement for The award of Bachelor of Technology in Computer Science and Engineering. It is a Bonafide report of the work carried out by us under the guidance and supervision of Mrs.E.Lalitha (Assistant Professor), Department of CSE-CYS,DS,AI&DS, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

Submitted by:

LINGALA BENJAMIN PAUL
21071A6232
CSE-CYBER SECURITY

KRISHNA NANDINI
21071A6231
CSE-CYBER SECURITY

ABSTRACT

Speech recognition is a fascinating technology that allows us to interact with our devices in a more natural and intuitive way. With the help of the Speech Recognition API in Python, we can convert audio input into text output, making it easier for us to communicate with our devices using our voices. This technology has been integrated into many popular voice assistants like Alexa, Siri, and Google Assistant, and it has become an essential component of the modern-day user experience.

One of the most common applications of speech recognition is in the domain of personal assistants. With the help of speech recognition, we can instruct our devices to perform a wide range of tasks like opening and closing applications, browsing the internet, playing music, setting reminders, and much more. These tasks can be performed using voice commands, making the experience more convenient and efficient for the user.

The process of converting audio into text using the Speech Recognition API in Python is relatively simple. We can use pre-built models that are available with the API or train our own models for specific use cases. Once the model is trained, we can pass the audio input to the model, which will then output the text. The output text can be further processed to perform the desired tasks based on the user's commands.

One of the biggest advantages of using speech recognition is that it eliminates the need for users to type in their commands manually. This makes the experience more accessible for people who may have difficulty typing, such as those with disabilities. Additionally, speech recognition can be used to improve the accuracy and speed of data entry, particularly in situations where the user needs to enter a large amount of data.

In the domain of personal assistants, speech recognition is particularly useful for performing tasks that require quick and accurate input. For example, a user might ask their personal assistant to play a specific song or artist, and the assistant can quickly retrieve the relevant information from the internet and play the music. Similarly, users can instruct their assistants to open and close applications or browse the internet for specific information, making it easier for them to multitask and be more productive.

Another key advantage of speech recognition is that it can be used to create more natural and intuitive user experiences. By allowing users to communicate with their devices using their voices, we can create more seamless and fluid interactions that feel less like using a computer and more like having a conversation with a friend. This can make the experience more engaging and enjoyable for users, and it can help to reduce the cognitive load required to perform complex tasks.

Table of Contents

Page No

1. INTRODUCTION-----	06
2. LITERATURE SURVEY-----	07
3. DESIGN-----	08
3.1. REQUIREMENT SPECIFICATION (S/W & H/W)-----	08
3.2. UML DIAGRAMS OR DFDs-----	09
4.CODE -----	10
5.IMPLEMENTATION-----	15
5.1MODULES-----	15
5.2OVERVIEW TECHNOLOGY-----	15
6.TESTING-----	16
6.1TEST CASES-----	16
6.2TEST RESULTS-----	16
7.RESULTS-----	17
8.CONCLUSION-----	18
9.OUTPUT-----	19
10.FUTURE SCOPE-----	20
BIBLIOGRAPHY-----	

1. INTRODUCTION

In the twenty-first century, technology has advanced at a rapid pace, leading to the development of new and innovative tools and devices that have revolutionized the way we live and work. One of the most significant advancements in recent times has been the development of virtual assistants.

A virtual assistant is a software program that can perform a variety of tasks on behalf of the user. These tasks can range from answering questions, setting reminders, scheduling appointments, sending emails, and making phone calls, among others. Virtual assistants are designed to make life easier for the user, by automating routine and time-consuming tasks.

One popular type of virtual assistant is the "Desktop Voice Assistant." As the name suggests, this type of virtual assistant is designed to work on desktop computers, and it uses speech recognition technology to understand and carry out the audio instructions given by the user.

The speech recognition technology used by desktop voice assistants is based on complex algorithms that can accurately convert spoken words into text. Once the text has been generated, the virtual assistant can use it to perform a variety of tasks, such as searching the internet, opening applications, and responding to emails.

One of the key benefits of using a desktop voice assistant is that it can save time and increase productivity. Instead of having to type out commands or search for information manually, users can simply speak their instructions, and the virtual assistant will do the rest. This can be particularly useful for people who work in fast-paced environments or who have a lot of tasks to complete in a short amount of time.

Another benefit of using a desktop voice assistant is that it can help to reduce the risk of repetitive strain injuries. Typing and using a mouse for long periods can be hard on the hands and wrists, leading to conditions such as carpal tunnel syndrome. By using a desktop voice assistant, users can reduce the amount of time they spend typing and clicking, which can help to prevent these types of injuries.

2.SYSTEM STUDY

Define the problem: This step involves defining the problem that the AI voice assistant is meant to solve. For example, the problem could be to provide hands-free control of smart home devices or to answer questions about a particular topic.

Requirements gathering: This step involves gathering information about what the AI voice assistant should be able to do, such as what commands it should recognize, what information it should provide, etc. This information should be gathered from the end-users, stakeholders, and other relevant parties.

Market research: This step involves researching existing AI voice assistants and their capabilities to determine what is currently available and what is lacking. This information can be used to guide the design and development of the AI voice assistant.

Technical feasibility: This step involves evaluating the technical feasibility of the AI voice assistant, including the availability of hardware and software resources, the level of expertise required to develop the AI voice assistant, and the potential costs involved.

Design and development: This step involves designing and developing the AI voice assistant. The design should take into account the requirements gathered in step 2 and the results of the market research and technical feasibility analysis.

Testing and evaluation: This step involves testing the AI voice assistant to ensure that it meets the requirements defined in step 2 and performs as expected. This step may involve testing the AI voice assistant in different environments, such as noisy environments or environments with poor acoustics, to ensure that it performs well in real-world situations.

Deployment: This step involves deploying the AI voice assistant to the end-users, who will use it to control smart home devices or to get information. **Maintenance and support:** This step involves providing ongoing support for the AI voice assistant, including bug fixes, updates, and other improvements as needed.

Overall, the goal of a system study for an AI voice assistant is to ensure that the AI voice assistant is designed and developed to meet the needs of the end-users, is technically feasible, and can be deployed and maintained effectively.

3.DESIGN

3.1. REQUIREMENT SPECIFICATION

SOFTWARE

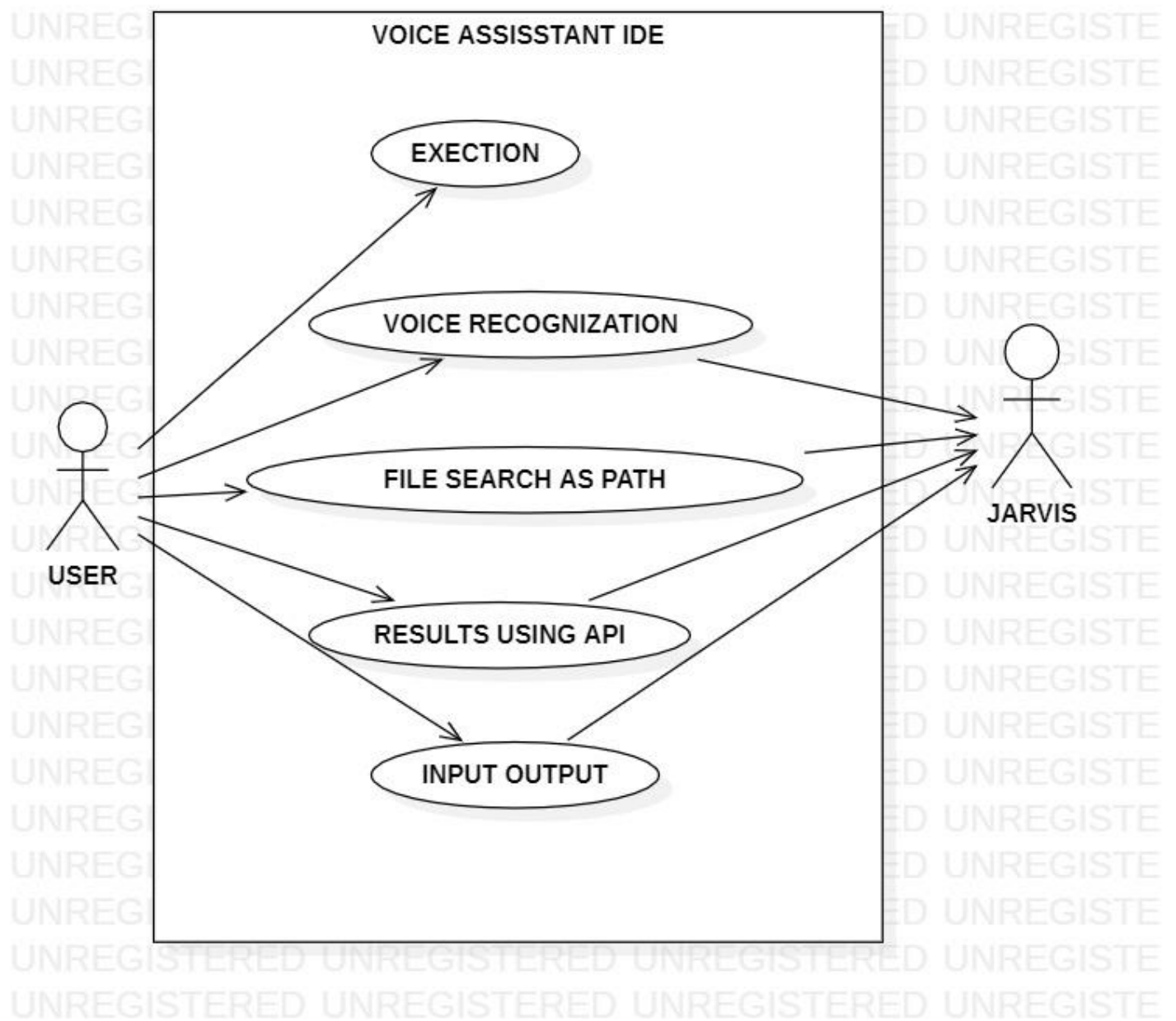
DESCRIPTION ARDUINO

IDE:

Figure 3.1 shows the Arduino Software IDE – It has a word processor for formulating code, a message area, a message console, a toolbar along with the options for normal capabilities and a sequence of menu [21]. It connects with the arduino apparatus to upload code into IoT boards and communicate with them. IDE is an open-source software, which is used to write and upload code to the Arduino boards Arduino. The IDE application is suitable for different operating systems such as Windows The, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment. The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino'[12][13].

Arduino Uno is an open-source microcontroller board developed by Arduino.cc. It is based on the Microchip ATmega328P microcontroller. It is one of the most popular Arduino development board and is universally known as 'stock Arduino'. It is a small development board having size 2.7 in * 2.1 in. It is both highly hardware and software compatible. This board is equipped with sets of digital and analog input/output (I/O) pins but the pin counts are less than Arduino, but Uno is good for small developmental project and prototyping. Its less cost and features makes it a good choice among engineers and students for project development [13].

3.2.USECASE DIAGRAM:



4.CODE:

```
#modules required
import pyttsx3
import json
import requests
import datetime
import speech_recognition as sr
import wikipedia
import webbrowser
import os
import random
import psutil
from win32com.client import Dispatch
from bs4 import BeautifulSoup

#To get the voice
engine=pyttsx3.init('sapi5')
voices=engine.getProperty('voices')
print(voices[0].id)
engine.setProperty('voice',voices[0].id)

#Commands to say while in process
lst_ans=['Command Accepted','Processing ma'am','okay ma'am','Your mish is my command']
lst_app=['My pleasure ma'am','It's my job ma'am','It makes me happy to help','I'm always here to help','Anytime ma'am','I am here to serve']

#To speak the assistant
def speak(audio):
    engine.say(audio)
    engine.runAndWait()

#Greetings before taking any command
```

```

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

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

    else:
        speak("Good Evening!")

    speak("I am Jarvis ma'am, Please tell me how may I help you")

#Taking command from the user in a voice
def takeCommand():
    # It takes microphone input from the user and returns string output
    r=sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1 #Time to wait before i complete saying energy_threshold
is another var to increse the voice of yours to give command
        audio=r.listen(source)
    try:
        print("Recognizing...")
        query=r.recognize_google(audio,language='en-in')
        print(f"user said : {query}")
    except Exception as e:
        print(e)
        speak("Say that again please...")
        return "None" #This is not a python none
    return query

#Closing all the opened tabs only
def close_app1(app_name):
    running_apps=psutil.process_iter(['pid','name']) #returns names of running
processes
    found=False

```

```

for app in running_apps:
    sys_app=app.info.get('name').split('.')[0].lower()

    if sys_app in app_name.split() or app_name in sys_app:
        pid=app.info.get('pid') #returns PID of the given app if found running

        try: #deleting the app if asked app is running.(It raises error for some
windows apps)
            app_pid = psutil.Process(pid)
            app_pid.terminate()
            found=True
            except: pass

        else: pass
    if not found:
        print(app_name+" not found running")
    else:
        print(app_name+'('+sys_app+')'+ ' closed')


#Newspaper reads for you
def speak1(string,i):
    speak=Dispatch("SAPI.spVoice")
    if num<10:
        speak.Speak(num)
    else:
        speak.Speak("And the last one is...")
    speak.Speak(string)


#main function starts here
if __name__=="main_":
    wishMe()
    while True:
        query = takeCommand().lower()

        speak(random.choice(lst_ans))

```

#Logic for executing tasks

#opening Wikipedia

if 'wikipedia' in query:

speak("Searching Wikipedia...")

query = query.replace("wikipedia", "")

results = wikipedia.summary(query, sentences=2)

speak("According to wikipedia")

print(results)

speak(results)

#opening youtube

elif 'open youtube' in query:

webbrowser.open("youtube.com")

#opening Google

elif 'open google' in query:

webbrowser.open("google.com")

#playing music from directory

elif 'play music' in query:

music_dir = "C:\\Users\\Win10\\Music\\" #c directory

songs = os.listdir(music_dir)

print(songs)

**os.startfile(os.path.join(music_dir,random.choice(songs))) #selecting
random songs from c directory**

#current time

elif 'the time' in query:

strTime = datetime.datetime.now().strftime("%H:%M:%S")

speak(f"Ma'am , the time is {strTime}")

#opening vs code

elif 'open code' in query:

**codePath = "C:\\Users\\Win10\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe" #It's a target of vs code not location of vscode**

os.startfile(codePath)

```
#closing Google if open  
elif 'close google' in query:  
    close_app1('chrome')
```

```
#closing youtube if open  
elif 'close youtube' in query:  
    close_app1('youtube')
```

```
#closing vs code if open  
elif 'close code' in query:  
    close_app1('code')
```

```
#close music file if open  
elif 'close music' in query:  
    close_app1('music')
```

```
#Checking current weather  
elif 'weather' in query:
```

```
    # enter city name  
    city = input("Enter the city name: ")
```

```
    # create url  
    url = "https://www.google.com/search?q="+ "weather"+city
```

```
    # requests instance  
    html = requests.get(url).content
```

```
    # getting raw data  
    soup = BeautifulSoup(html, 'html.parser')  
    temp = soup.find('div', attrs={'class': 'BNeawe iBp4i AP7Wnd'}).text
```

```
    # this contains time and sky description  
    str = soup.find('div', attrs={'class': 'BNeawe tAd8D AP7Wnd'}).text
```

5.IMPLEMENTATION

1. Subprocess
2. Wolfram Alpha
3. Pyttsx3
4. Tkinter
5. Wikipedia
6. Speech recognition
7. Web Browser
8. Ecapture
9. Pyjokes
10. Datetime
11. Twilio
12. Requests
13. BeautifulSoup

- ✓ **Subprocess** - used to run new applications or programs through Python code by creating new processes. It also helps to obtain the input/output/error pipes as well as the exit codes of various commands.
- ✓ **WolframAlpha** - an API that can calculate expert-level answers with the help of the algorithms, knowledgebase, and Artificial Intelligence (AI) technology of Wolfram which is made possible by the Wolfram Language.
- ✓ **Pyttsx3** - a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.
- ✓ **Tkinter** - It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications.
- ✓ **Wikipedia** - This module allows us to get and parse the information from Wikipedia. In simple words, we can say that it is worked as a little scrapper and can scrap only a limited amount of data.
- ✓ **Speechrecognition** - 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.
- ✓ **Web Browser** - a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users. webbrowser can also be used as a CLI tool.
- ✓ **Ecapture** - This module is used to capture images from your camera.
- ✓ **Pyjokes** - a python library that is used to create one-line jokes for programmers.
- ✓ **Datetime** - supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals.
- ✓ **Twilio** - a web application programming interface (API) that software developers can use to add communications such as phone calling, messaging, video and two-factor authentication into their Python applications.
- ✓ **Requests** - allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).
- ✓ **BeautifulSoup** - a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

6. TESTING

Testing is an important part of the development process for AI voice assistants. Here are some of the types of testing that are commonly performed on AI voice assistants:

Functionality testing: This type of testing involves checking that the AI voice assistant can perform the tasks it was designed to perform, such as recognizing and responding to voice commands, providing information, and controlling smart home devices.

User interface testing: This type of testing involves checking that the user interface of the AI voice assistant is intuitive and easy to use, and that it provides clear and useful feedback to the user.

Usability testing: This type of testing involves evaluating the AI voice assistant from the perspective of the end-user to see how well it meets their needs and how easy it is to use. This type of testing can be performed through surveys, focus groups, or user testing sessions.

Performance testing: This type of testing involves evaluating the performance of the AI voice assistant, such as its response time, processing speed, and resource usage. This testing is important to ensure that the AI voice assistant performs well even under heavy load.

Security testing: This type of testing involves evaluating the security of the AI voice assistant, including its ability to protect sensitive information and to resist attacks from malicious actors.

Compatibility testing: This type of testing involves checking that the AI voice assistant works well with other devices, such as smart home devices, wearable technology, and mobile devices.

Regression testing: This type of testing involves checking that changes or updates to the AI voice assistant have not negatively impacted its performance or functionality.

Overall, the goal of testing for an AI voice assistant is to ensure that it is reliable, efficient, and easy to use, and that it meets the needs of the end-users. Testing is an ongoing process that should be performed throughout the development life cycle of the AI voice assistant.

7.RESULTS

There are 6 python files used to develop our Desktop Voice Assistant.

1. actions.py
Modules included - configparser, datetime, webbrowser, pytsx3, requests.
This module is to setup your Desktop Voice Assistant configuration.
2. commands.py
Modules included – configparser, random, smtplib, sys, wikipedia, mixer from pygame.
And also from actions we included open_url, search and speak.
This file processes your speech and give output according to it.
3. gui.py
Modules included - tkinter
This file makes GUI of our Desktop Voice Assistant.
4. p.py
Modules included - configparser
For setting the configuration to default.
5. Jarvis2_4windows.py
Modules included – configparser, os, gui, actions, commands, speech_recognition.
This file is for user queries. It takes commands from user and identify the desired task that user wants to perform and generate output according to it.
6. Jarvis2.py
Modules included - datetime, getpass, os, random, smtplib, sys, webbrowser, pytsx3, speech_recognition, wikipedia, gui.

This is a file which opens interface to Desktop Voice Assistant where you can meet your Voice assistant and chat with it.

So by using all these modules together we tried to built our own Desktop Voice Assistant .
This Voice Assistant performs basic operations with no difficulty.

8.CONCLUSION

Desktop Voice Assistants (VPAs) have become an essential tool for organizing schedules and managing tasks. They offer many benefits over human personal assistants, including their portability, 24/7 availability, and access to vast amounts of information. With the help of artificial intelligence and the Internet of Things, these devices are becoming even more powerful and useful for users.

One of the key advantages of VPAs is their ability to use artificial intelligence to learn from user behaviour and improve over time. This means that as users interact with their VPA, the software can learn their preferences and adapt to their needs, making it more effective at performing tasks and managing schedules. As the software becomes more sophisticated, it can also anticipate user needs and proactively offer suggestions and solutions, further enhancing its usefulness.

Another important benefit of VPAs is their ability to connect with other devices through the Internet of Things. This means that users can use their VPA to control smart home devices, such as lights, thermostats, and security systems, without ever leaving their desk or picking up their phone. By integrating with other devices in the home or office, the VPA can provide a seamless and intuitive experience for users, making it easier to manage their environment and increase their productivity.

Finally, VPAs are becoming more powerful and versatile with the increasing availability of data and the development of advanced algorithms. By analyzing vast amounts of information and using machine learning to identify patterns and trends, VPAs can provide insights and recommendations that can help users make better decisions and achieve their goals more efficiently.

9.OUTPUT:



10.FUTURE SCOPE

The future of AI voice assistants is bright and holds great potential. Here are some of the areas where AI voice assistants are likely to see significant growth and development in the coming years:

Improved natural language processing: AI voice assistants are likely to become even better at understanding and responding to human speech, allowing for more natural and intuitive interactions.

Integration with other technologies: AI voice assistants are likely to be integrated with other technologies, such as smart home devices, wearable technology, and autonomous vehicles, to provide even more functionality and convenience.

Customization and personalization: AI voice assistants are likely to become more personalized, allowing users to customize the assistant to meet their specific needs and preferences.

Increased security and privacy: As AI voice assistants become more widespread, there will be a greater focus on ensuring the security and privacy of the data that is collected and processed by these assistants.

Expansion into new markets: AI voice assistants are likely to expand into new markets, such as healthcare and education, providing new opportunities for innovation and growth. **Increased automation:** AI voice assistants are likely to become more automated, allowing users to perform complex tasks with just a few simple voice commands.

Development of more advanced AI models: As AI technology continues to advance, AI voice assistants are likely to become even more sophisticated, allowing for more accurate and helpful interactions.

Overall, the future of AI voice assistants is exciting and holds great potential for improving our lives and making our world a better place.