VOICE ASSISTANT: DESKTOP-BASED APPLICATION

Jay Mhatre[1], Prasanna Tayare[2], Sanjeev Kumar[3], Pratham Temkar[4], Dr. Mahendra Pawar[5]

1. Student, Dept. of COMP Engineering, VPPCOE & VA, Maharashtra, India
2. Student, Dept. of COMP Engineering, VPPCOE & VA, Maharashtra, India
3. Student, Dept. of COMP Engineering, VPPCOE & VA, Maharashtra, India
4. Student, Dept. of COMP Engineering, VPPCOE & VA, Maharashtra, India
5. Associate Professor, Dept. of COMP Engineering, VPPCOE & VA, Maharashtra, India

---***---

Abstract - *With voice assistants emerging as essential elements of advanced computer systems, they provide a convenient and user-friendly way to interact with the technology. This research project concentrates on creating and using an Electron JS and Python-based voice assistant that is desktop-oriented. The work investigates technical architecture, development process, and user experience aspects associated with designing a desktop-based interface driven by voice. Using Electron JS allows for compatibility across platforms as well as incorporating web technologies while Python provides core features like speech recognition and natural language processing. This project reveals some aspects of creating a voice assistant for desktop use such as speech accuracy, UI design, system integration, etc. Feedback from users and testing results help shed light on the usability and efficiency of the voice assistant application. Finally, this paper discusses possible future directions for improvements to make these types of tools more widely applicable in other areas beyond desktop assistance. The contribution made by this study is focused on how to advance the field of voice assistant software toward being integrated into desktop environments.*

Key Words: *Voice Assistant, Recognition, NLP, Desktop*

1. INTRODUCTION

In recent years, voice assistants have revolutionized how we interact with technology, providing a convenient and natural way for people to engage with their devices. Today, these assistants have become an integral part of our daily lives, assisting with everything from reminders to controlling smart home devices. While voice assistants have traditionally been linked to portable and home-based electronics, there is a growing demand to expand their functionality to desktop computer settings. This research project aims to create and deploy a desktop voice assistant application using Python and Electron JS. The proposed voice assistant seeks to enhance accessibility and productivity by enabling hands-free interaction with desktop computers. Unlike traditional voice assistants that are primarily made for mobile or smart speaker platforms, this project will use web technologies like HTML, CSS, and JavaScript to develop cross-platform desktop applications using Electron JS. Our objective is to develop a voice assistant that is optimized for desktop use, with a focus on flexibility and user-friendliness. We will leverage Python's powerful libraries and frameworks for natural language processing and speech recognition to achieve this goal. Our primary aim is to create an automated personal assistant that enhances user productivity and convenience while demonstrating the feasibility of this technology. Our prototype will showcase basic functionalities, laying the groundwork for future advancements and optimizations.

2. BACKGROUND

2.1. History of Voice Assistant

From simple speech recognition software to advanced AI-powered virtual assistants, voice assistants have come a long way. The voice assistant journey, however; had started in the mid-20th century with crude attempts at speech recognition. By the 1950s Bell Labs had introduced “Audrey” then by the 1960s IBM came up with “Shoebox”. These served as a basis for further development. Nonetheless, these earliest systems were limited by technology and failed to gain widespread use due to their inadequacies and limited accuracy. This took place at the turn of the century when more advanced voice assistants were being developed. As early as 2001, Microsoft’s “Cortana” was launched, which led to Apple’s release of Siri on iPhone 4S in 2011. Introduced via iPhone 4S, Siri ensured that everyone bought into the virtual assistant concept through natural language command hence starting an era of voice-controlled interface. These are among some of the voice assistant platforms that have dominated the market such as Amazon’s Alexa in 2014 and Google Assistant released in 2016. These platforms used artificial intelligence (AI), natural language processing (NLP), and cloud computing so that they became increasingly personalized and responsive. Today there are voice assistants implemented across various devices or services thereby transforming how users interact with technology every day.

2.2 Future Applications

Some other future applications of voice assistants are personalized healthcare support, advanced residential automation with predictive capabilities, immersive virtual reality experiences with natural language interaction, and enhanced accessibility features for disabled people. By this means, the voice assistant will become more embedded in daily life, offering convenience and help across a range of contexts.

2.3 Aim of this Study

Our research aims to pave the way for more efficient and natural human-computer interactions by integrating voice assistant technology into desktop computing environments.

3. PROPOSED DESIGN

The project will provide a reasonable grasp of an intelligent assistant that can comprehend user commands. Our voice assistant recognizes and complies with commands given by the user through vocal media and responds as required. We develop a prototype of an automated personal assistant capable of understanding user commands and performing tasks across various domains, including scheduling, information retrieval, and task automation. Using web technologies, the proposed desktop voice assistant application will be built cross-platform for desktop apps using the Electron JS framework. Using HTML, CSS, and JavaScript, Electron offers a single development environment that makes it easy to integrate with current web-based technologies while developing desktop apps. Python will be used to implement the application's backend features, which include speech recognition and natural language processing. Python provides strong frameworks and tools to perform these tasks such as pyttsx3, SpeechRecognition, Pyaudio, etc.

Our study aims to achieve several important goals, including:

- To investigate the principles of design and technical architecture of desktop voice assistants.
- To look at the difficulties and development process involved in creating a voice-activated desktop interface.
- To investigate the factors related to user experience and usability when utilizing a desktop voice assistant.
- To test and assess the voice assistant application to obtain user input and insights.
- To determine future paths and possible improvements for desktop voice assistant technology.

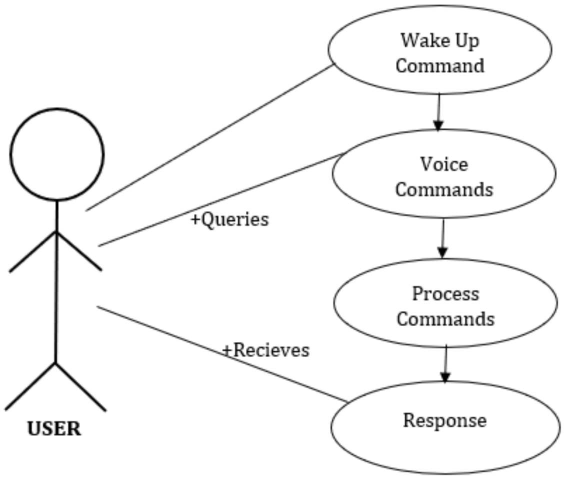


Figure 1: Use-Case Diagram

In this diagram, labeled as Figure 1, we can see the Use-Case illustration showcasing the interaction between the user and the voice assistant. The user initiates the conversation by speaking to the assistant, who then proceeds to identify their speech, interpret their intentions, carry out relevant tasks, and generate a fitting response. The assistant has to furnish the requested information or perform the necessary action.

4. METHODOLOGY

When a user accesses a Voice Assistant application, they are presented with a desktop application that allows them to interact with the platform. The backend functionalities of the assistant were built using Python, while web technologies such as HTML, CSS, and JavaScript were used to create the frontend interface through Electron JS. To enable communication between the frontend and backend components, FastAPI was implemented. Various modules were developed and integrated using specific technologies and frameworks to support the Voice Assistant's basic operations, AI, voice recognition, face recognition, emotion identification, and system application control. To enable system voice within the program, we utilized Sapi5 and Pyttsx3. The latter is a Python text-to-speech conversion module that operates offline. Additionally, we incorporated the Speech Recognition Library, which provides numerous built-in functions. This library allows users to send commands to the assistant and receive voice responses in return, complete with Text-to-speech capabilities. Upon recording, the assistant's underlying algorithms convert the user's voice command to text. When a user gives a command such as "Open Notepad" or "Play a Song on YouTube", the voice assistant responds by executing the command. The assistant waits for a pause to indicate that the user has finished speaking, then searches its database for the user's command and executes it accordingly.

- The user's request is split into separate commands to aid understanding by the voice assistant.
- Once the command list is accessed, it searches and compares our requests with others.
- The list of commands is then sent back to the voice assistant.
- After receiving the commands, the voice assistant will know what action to take next.
- The voice assistant will ask for clarification if a request is unclear to ensure it understands the user's desired outcome.
- If the voice assistant can understand the user's request, it will perform the task.

In the final phase of distribution, the Voice Assistant was deployed to desktop environments while adhering to operating system standards and ensuring compatibility with various platforms.

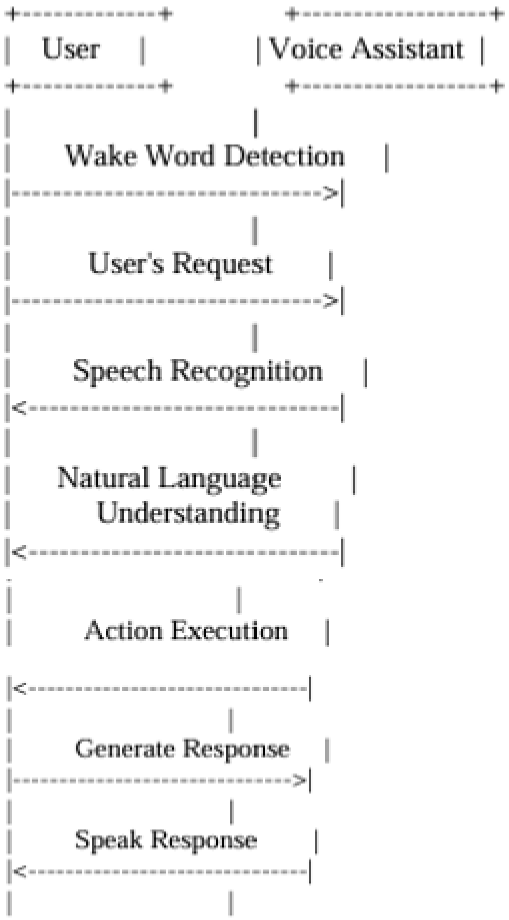


Figure 2: Sequence Diagram

Figure 2 shows the sequence diagram that displays the interaction between the user and the voice assistant. The assistant wakes up on the wake word and listens to the user's request. The request is converted to text, analyzed, and mapped to an action. The assistant executes the action and generates a response, which is spoken to the user.

5. IMPLEMENTATION & RESULTS

"Sifra," the voice assistant program for desktop use, is designed with an intuitive user interface. When the application is launched, users will be presented with a login page that utilizes facial recognition for authentication. This system is crucial in ensuring user identification and secure access to the Assistant interface. Authenticating user credentials, limits access to only authorized users who can interact with the Assistant and execute commands. Ultimately, this feature safeguards user privacy and provides a trusted and secure interface.

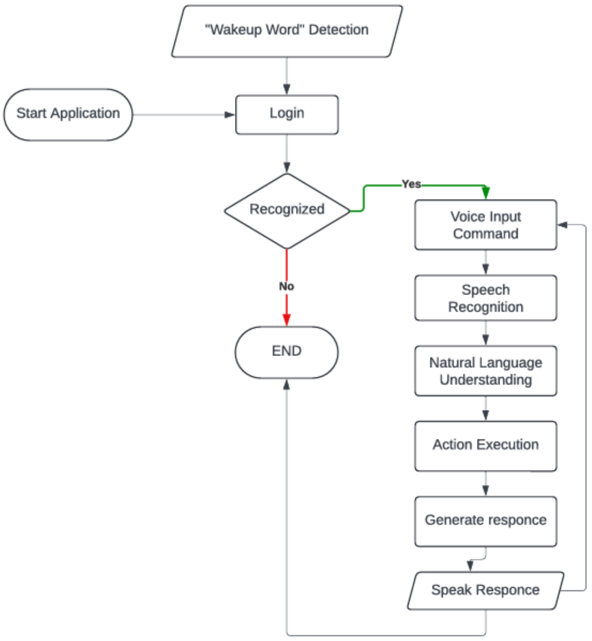


Figure 3: Flow Chart

Figure 3 shows a flowchart where a user activates the voice assistant using a "wake-up" keyword. When the voice assistant is activated, it will greet the user and wait for the user to provide an input command. If the user provides an input command through voice, the assistant will capture it and search for any related keywords. If the assistant finds a keyword related to the input command, it will perform the task and return the output to the user both in voice and in text form on the terminal window. If the assistant cannot find any relevant keywords, it will continue waiting for the user to input a valid command. Each of these functionalities plays an important role in the overall functioning of the system.

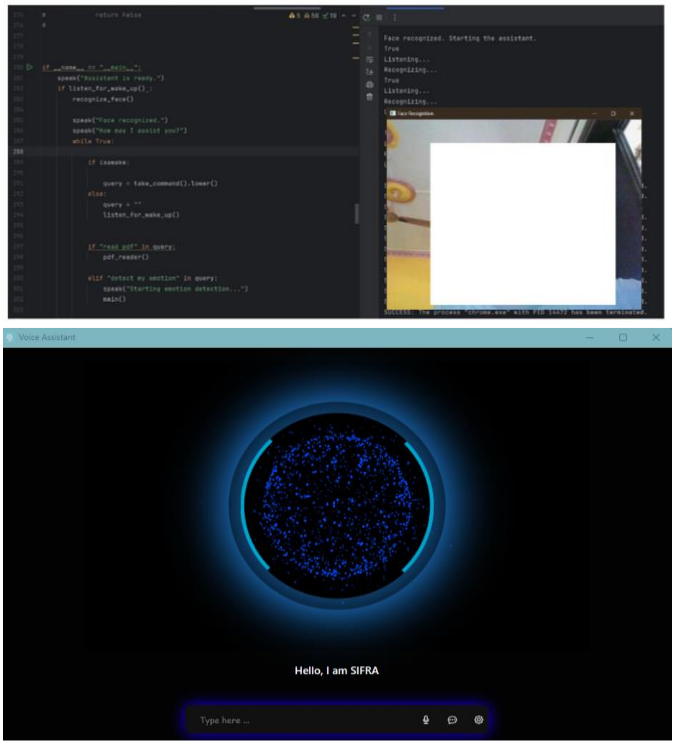
The following tasks are performed by the Voice Assistant:

- Launching applications, setting reminders, creating calendar events, sending messages, etc.
- Searches anything from Google and tells the required content. If asked 'Google search' the

assistant searches the content asked from Google and opens the required content in the browser.

- Tells the accurate weather of the location the user asks for. When asked for 'current weather in' the assistant tells the exact weather of the desired location of both maximum and minimum in degrees Celsius.
- Takes a screenshot of the display. When asked for "capture", "capture my screen", "my screen", "screenshot", or "take screenshot", the assistant captures the display the user is using and stores it in the path specified.
- It sends a mail to the username specified by the user. When told "send mail", the assistant asks to whom the mail has to be sent and it will send a mail according to that.
- Can shut down or restart the system by just user command etc.

Login through face recognition and activation of the assistant:

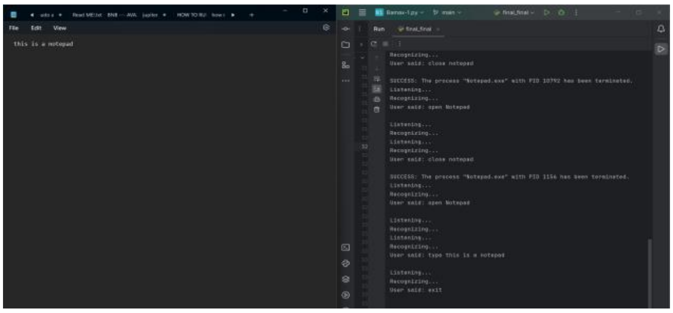


```
if __name__ == "__main__":
    speak("Assistant is ready.")
    if listen_for_wake_up() :
        recognize_face()

        speak("Face recognized.")
        speak("How may I assist you?")
        while True:

            if iswake:

                query = take_command().lower()
            else:
                query = ""
                listen_for_wake_up()


            if "read pdf" in query:
                pdf_reader()

            elif "detect my emotion" in query:
                speak("Starting emotion detection...")
                main()
```

```
Face recognized. Starting the assistant.
True
Listening...
Recognizing...
True
Listening...
Recognizing...
SUCCESS: The process "chrome.exe" with PID 14472 has been terminated.
```

Figure 4: Opening Desktop Application

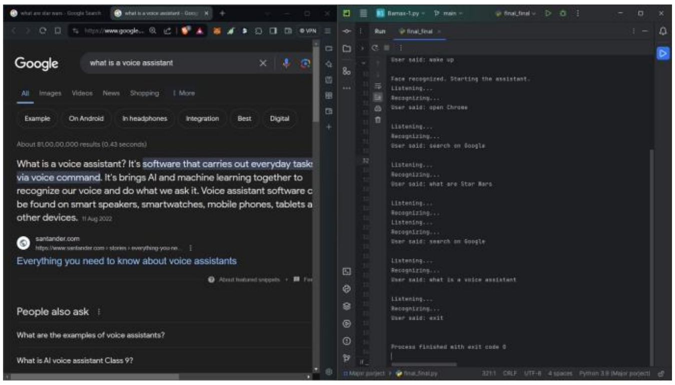
Assigning Task To the Assistant:



this is a notepad

```
Recognizing...
User said: close notepad

SUCCESS: The process "Notepad.exe" with PID 10792 has been terminated.
Listening...
Recognizing...
User said: open Notepad

Listening...
Recognizing...
Listening...
Recognizing...
User said: close notepad

SUCCESS: The process "Notepad.exe" with PID 1156 has been terminated.
Listening...
Recognizing...
User said: open Notepad

Listening...
Recognizing...
Listening...
Recognizing...
User said: type this is a notepad

Listening...
Recognizing...
User said: exit
```

Figure 5: Asking the assistant to open Notepad



what is a voice assistant

What is a voice assistant? It's software that carries out everyday task via voice command. It's brings AI and machine learning together to recognize our voice and do what we ask it. Voice assistant software c be found on smart speakers, smartwatches, mobile phones, tablets a other devices. 11 Aug 2022

Everything you need to know about voice assistants

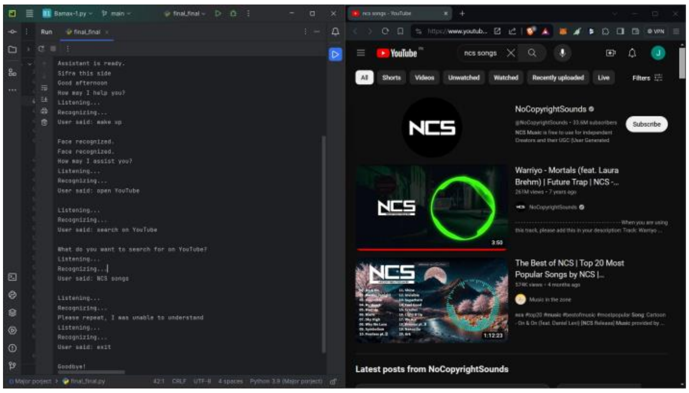
People also ask

What are the examples of voice assistants?

What is AI voice assistant Class 9?

```
User said: wake up

Face recognized. Starting the assistant.
Listening...
Recognizing...
User said: open Chrome

Listening...
Recognizing...
User said: search on Google

Listening...
Recognizing...
User said: what are Star Wars

Listening...
Recognizing...
Listening...
Recognizing...
User said: search on Google

Listening...
Recognizing...
User said: what is a voice assistant

Listening...
Recognizing...
User said: exit


Process finished with exit code 0
```

Figure 6: Asking the assistant to search on Google



```
Assistant is ready.
Sifra this side
Good afternoon
How may I help you?
Listening...
Recognizing...
User said: wake up

Face recognized.
Face recognized.
How may I assist you?
Listening...
Recognizing...
User said: open YouTube

Listening...
Recognizing...
User said: search on YouTube

What do you want to search for on YouTube?
Listening...
Recognizing...
User said: NCS songs

Listening...
Recognizing...
Please repeat, I was unable to understand
Listening...
Recognizing...
User said: exit

Goodbye!
```

Ramax-1.py
main
final_final
Run
YouTube
ncs songs
All
Shorts
Videos
Unwatched
Watched
Recently uploaded
Live
Filters
NoCopyrightSounds
Subscribe
Warriyo - Mortals (feat. Laura Brehm) | Future Trap | NCS
The Best of NCS | Top 20 Most Popular Songs by NCS |...
Latest posts from NoCopyrightSounds

Figure 7: Asking the assistant to play music on YouTube

6. CONCLUSION

In summary, making the Voice Assistant app for desktops was a big effort to make computer use better with new tech. Using Electron JS and Python, the project made a voice assistant that can do voice, face, and emotion recognition, control system apps, do basic tasks, and support artificial intelligence. By following a plan that involved finding needs, doing tech research, designing, making, testing, and writing stuff down, the project did what it set out to do by giving users a smooth and easy time. In the future, the Voice Assistant application has vast potential for further advancements and applications across multiple domains. It could understand language better, do more things, and work with new tech. Overall, the Voice Helper project is a big step forward in how people and computers work together, making it easier and more personal to use computers.

ACKNOWLEDGEMENT

We would like to sincerely thank Dr. Mahendra Pawar, our project guide, and the rest of the computer office's teaching team for their significant suggestions, encouragement, and support during this research. Their knowledge and perceptions have been crucial in determining the course and extent of this undertaking. We also acknowledge the Department Head, Dr. Rais Mulla, for his unwavering encouragement and assistance. His leadership has served as an inspiration to us. We also like to thank our peers and colleagues for their contributions, as they have offered insightful criticism and recommendations. Their advice has substantially improved our job.

REFERENCES

[1] Subhash S, Prajwal N Srivatsa, Siddesh S, Ullas A, & Santosh B "Artificial Intelligence-based Voic-Assistant" IEEE 2020.

[2] Ms. Preethi G, Mr. Thiruppugal, Mr. Abhishek, & Mr. Vishwas D A "Voice Assistant using Artificial Intelligence " IJERT 2022.

[3] G Rushivardhan, & Mrs. K Santoshi "Virtual Assistant using NLP Techniques" IJCRT 2022.

[4] Veton Kepuska, & Gamal Bohouta "Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Home) " IEEE 2018.

[5] G. Bohouta and V. Z Kepuska "Improving Wake-up-Word and General Speech Recognition Systems". IEEE 2017 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress.

[6] Ajay Kumar Sahul, Shivani Dubey, Ashish Kumar Jha, Ritik Bhargava, Priyanshu Priya, Rupa Kumari "Voice Assistant Using Artificial Intelligence"

[7] Nallamothu M., & Mukkamala R. "A study on Speech Recognition and Desktop Assistant using Python" Journal of Emerging Technologies and Innovative Research 2019.

[8] A N Bhoomika Chowdary, Lavanya M, Darshan S, Deepthi Andani, & Mrs. Jayasudha BSK "A Literature Survey on Voice Assistant" IJRASET 2022.

[9] Faruk Lawal Ibrahim Dutsinma, Debajyoti Pal, Suree Funikul, & Jonathan H. Chan "A Systematic Review of Voice Assistant Usability: An ISO 9241-11 Approach" 2022.

[10] S. Springenberg, M. Moberg, & O. Viikki, "Intelligent Personal Assistants" Speech Technology Seminar, Institute of Computer Science Hamburg, 2016.

[11] V. Kukade, Ruchita G. Fengse, Kiran D. Rodge, Siddhi P. Ransing, & Vina M. Lomte, "Virtual Personal Assistant for the Blind", International Journal of Computer Science and Technology, Volume 9, Issue 4, October-December 2018.

[12] Ritik Porwal, Ujjawal Tomar, Vishakha Dubey, Asst. Prof. Akshita Mishra, Asst. Prof. Gourav Mandloi "Voice Assistant" ISSN 2021.