



ISSN: 2454-9940



**INTERNATIONAL JOURNAL OF APPLIED
SCIENCE ENGINEERING AND MANAGEMENT**

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

PYTHON POWERED PERSONAL VIRTUAL ASSISTANT

¹ Ameena Farheena, ² Nazia Tazeen, ³ Mohammed Waheeduddin Hussain

¹ PG Scholar, Department of Information Technology, Shadan College of Engineering and Technology,
Hyderabad, Telangana, India – 500086. ameenafarheena14@gmail.com

² Professor, Department of Information Technology, Shadan College of Engineering and Technology,
Hyderabad, Telangana, India – 500086. naziabaseer@gmail.com

³ Professor, Department of Information Technology, Shadan College of Engineering and Technology,
Hyderabad, Telangana, India – 500086. mwaheeduddinhussain@gmail.com

Abstract: In the advancing domain of intelligent systems, assistants with the support of voice example of pragmatic use of artificial intelligence (AI) are to improve the interaction with the human computer. Personal virtual assistant based on desktop computers has been developed using Python to perform tasks at the level of system and the Internet using natural voice commands. The assistant uses voice recognition to transform the audio input into a machine-readable text and uses the synthesis of the text to the speech to provide answers, facilitating the Hands-free conversation. Basic libraries include speech_recognition, Pyttsx3, OS and Web browser, enabling voice processing, execution of commands and automating tasks. The system design facilitates modular and offline functionality, increases privacy and reduces cloud services. Task performing include access to websites, extract information from Wikipedia, playing local media files and presentation of time -related statistics. This approach, unlike commercial voice assistants depending on web services and proprietary frames, guarantees user control, reduced use of resources and flexibility of adaptation. The performance study reveals the average accuracy of 70% identification and reaction time of 2 to 4 seconds, indicating practical use on conventional table hardware. The project demonstrates the integration of the AI speech interface with desktop computer environments to increase availability, minimize input latency and optimize users' involvement.

Index Terms - Voice Assistant, Speech Recognition, Text-to-Speech, Artificial Intelligence, Python, Desktop Automation, Natural Language Processing, Offline Functionality.

1. INTRODUCTION

Currently controlled technology transition to automation and intelligent interaction revolutionized users' connection with computer systems. Virtual assistants with voice support have become important technologies that facilitate digital

interactions by allowing users to perform activities using natural language requirements. These assistants use development in artificial intelligence (AI), speech recognition and natural language processing to provide smooth communication between people and machines. [1] [2]. Commercial systems such as Apple Siri, Amazon Alexa and

Microsoft's Cortana illustrate this trend by providing help in online search, reminders, playing media and system management [5].

Regardless of their widespread attraction, most commercial voice assistants rely on cloud infrastructure and are adapted to mobile or intelligent domestic ecosystems and provides limited offline functionality and adaptation for desktop settings [4] [6]. In addition, personal data protection problems and limited extensibility prevent their use in sensitive or personalized computing environments [3] [7]. To alleviate these restrictions, a voice assistant for desktop computers from Python, called Lisa, was created. This system operates using the "Lisa" voice commands, allowing Hands-Free to perform activities such as Wikipedia search, access to files, playing music and getting information in real time. The assistant employs Python packages such as speech_recognition, PytSX3, OS, Webbrowser and DateTime to provide voice interaction, speech synthesis and automate commands [8] [9].

Lisa, which worked mostly offline for local operations, guarantees privacy and efficiency and offers efficient solutions to automate desktops. Modular architecture makes it easier to adapt and insert new features, allowing progressive expansion of capabilities over time. Individuals with accessibility requirements or those aimed at increasing productivity can gain considerable benefits from this technology, especially in educational or personal computer contexts.

The aim of this system is to provide an intelligent, voice -activated assistant that improves desktop computers through automation controlled AI. The assistant illustrates the practicality and importance

of the natural language interface in daily calculations by integrating python programming with voice technology with open source code. [1] [6].

2. RELATED WORK

Recent breakthroughs in artificial intelligence and speech catalyzed the spread of virtual assistants with voice support for desktop and mobile platforms. These systems seek to automate secular work and improve the availability of users by processing voice commands of natural language. Numerous academics examined the deployment of voice - controlled assistants using Python and AI frames, recognizing their growing importance in the area of interaction with the human computer.

The remarkable contribution in this domain is provided by Garg and Verma [10], who created a virtual assistant based in Python, which causes basic operations on the desktop. Their technology combined voice recognition with commands to perform activities such as running applications, obtaining websites, and providing information in real time. They set up the viability of the use of Python Open-Source Libraries for Offline Voice Interactions, making their assistant suitable for personal computing settings with limited Internet connection.

Abraham et al. [11] He made a similar attempt by introducing EVA, a virtual assistant on a desktop computer based in Python adapted to visually questioned individuals. Their approach emphasized the accessibility and minimal use of resources. The assistant worked in offline mode by means of voice recognition for input synthesis and text on output speech. Unlike commercial voice assistants who

often threaten privacy, Eva implemented local data processing, reducing reliance on cloud services and increasing control of personal data. Modular design of the system allowed developers to include additional features for wider use.

Pradhan et al. [12] They examined the user's expectations and requirements in the Nova Virtual Assist Studio. Their research integrated user input and behavior analysis to create a personalized assistant capable of adapting to individual preferences. The study emphasized the importance of context systems and the need for customizability in adapting responses and tasks. The NOVA system functioned as a prototype to improve the quality of life through voice interaction and digital automation, especially in residential and workplaces.

MANE et al. [13] He introduced a light virtual personal assistant for desktop computers developed in Python and emphasized basic activities such as browsing, media control and file management. Their methodology focused on reducing relying on advanced equipment and internet connection, which is particularly beneficial for educational and rural contexts. Scientists have emphasized the advantages of integrating existing Python modules for optimized development and efficient use of resources and advocate access to offline first.

Subi et al. [14] In addition, examine the progress of artificial intelligence -based assistants by introducing the desktop, see, a top -of -the -line activated assistant. Their aim was to amalg to sustainability with AI by allowing an assistant to run with minimal hardware restrictions in performing tasks such as reading text, reminders of tasks, and media management. The system used the synthesis and recognition of voice to minimize the screen

relocation, especially for seniors and those with visual impairment. This emphasized the growing importance of inclusive design in voice systems.

Rao et al. [15] They created a voice virtual personal assistant aimed at strengthening the interaction with the design of the desktop. Their assistant could automate the voice input instructions such as opening directory, starting browsers, and loading time and data. This study has shown that rules -based voice assistants can function efficiently without the need for deep learning or complex machine learning models, reducing the system's computing load.

Archite and Saravanan [16] examined the educational consequences of customizable voice assistants and created a virtual assistant to strengthen students' learning. Their system included planning controlled commands, notifications and resolution of queries. The assistant facilitated voice orders for educational aid functions, increased learning efficiency and reduced digital abyss. Their methodology emphasized the adaptability of intelligent systems to instructional contexts and specific requirements of individual students.

Kumar et al. [17] He introduced Ziva-AI, a virtual assistant for Windows. The assistant used voice recognition and the processing of natural language to perform tasks at the system level. He had the ability to navigate, search files and automate tasks. Scientists emphasized real -time sensitivity and intuitive interface of speech and showed the potential of open source code technology to create a highly engaging desktop experience.

Rhardian et al. [18] He provided a prototype for artificial intelligence integrated virtual assistant

system. Their design preferences flexibility and expandability and adapted different instructions and activities of the user. The assistant provided voice processing, Internet access, and local file management, and showed how the basic AI integration can improve users' connection and system performance in several sectors.

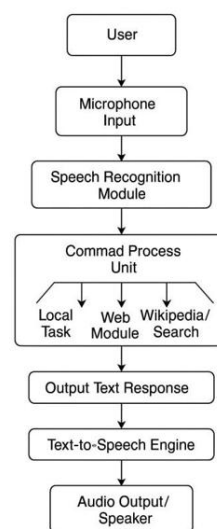
Benny et al. [19] They examined the use of Openai technology to develop a personal assistant for desktop computers. Their assistant used the processing of GPT -based text and dynamic commands to provide more intelligent conversational interactions. Using the Openai models, the assistant can manage more complex inputs and provide context -relevant answers. Their research has illustrated how the current LLM can be integrated into desktop settings to transform the interaction and efficiency of users.

In the wider frame of AI Karn et al. [20] They examined the development and deployment of various AI assistants and emphasized their potential in many industries such as health care, education and consumer applications. Their results emphasized the need to develop intelligent systems that are adaptive, efficient sources and proficient in increasing availability through voice interaction.

Together, these experiments illustrate the considerable potential of voice assistants based on Python in meeting the real world requirements, especially in desktop environments. Each assistant, whether intended for availability, personalization or improve education, increases the development of intelligent systems focused on users that combine human purpose with digital implementation.

3. MATERIALS AND METHODS

This study introduces light, offline capable and modular voice assistant called Lisa, created by Python and AI frames, to solve shortcomings of commercial voice assistants, including relying on cloud infrastructure, personal data protection problems and insufficient desktop optimization. The system detects the specified Wake Word, analyzes the user's voice input by speech_recognition, and provides immediate responses through the PytSX3 speech synthesis. Unlike conventional assistants, LISA performs most operations locally - for example, starting applications, getting date and time and accessing local files - privacy and reducing resource use [11]. Modular assistant design for direct adaptation and integration of other abilities, such reminders, access to e-mail or API-based application [13]. In addition, it helps individuals with engine damage or those who require hand-free control, which is exceptionally accessible to the domestic or educational environment [14]. This method illustrates the use of Python and Open-Source technologies to develop scalable, efficient and secure desktop assistants for everyday applications.



“Fig 1 System Architecture”

The image "Fig. 1 System Architecture" shows the operation of the voice system. The user supplies the microphone input that is analyzed by the speech recognition module. The result is then sent to a command process unit that could assign a local task request, web module or Wikipedia/search. The output text response is finally created, converted into a sound using the engine to speech and is presented to the user as an audio output/speaker.

i) Voice Input and Wake Word Activation

The system microphone constantly monitors the voice input via the speech_recognition module. The awakening detection system, set to the "press", activates the assistant listening mode for other requirements. To improve accuracy of recognition, the control of the surrounding noise and normalization of input is used. After Wake Word detects, Lisa records another voice and rewrites it into the text using the Google Voice Voice Voice. Although the Internet connection is necessary to recognize the voice, all other functions are performed locally. The assistant offers real-time console logging to increase the usability for tuning purposes. The voice activation and listening mechanism is designed to be effective, fast and suitable for offline settings, where a constant cloud connection may miss. This methodology is influenced by the assistant Eva created by Abraham et al. [11], which favors the recognition of the word, local processing and accessibility for people with disabilities.

ii) Command Processing and Task Execution

After converting the voice into text text, the processing of the commands begins with aligning keywords in the input with the instruction set. For example, if this term includes "YouTube", Lisa initiates YouTube using the Webbrowser module. OS and sub-modules are used for system level operations such as file access and audio playback. The system uses logic based on rules, instead of machine learning models, IF-ELSE or MATCH-CASE structures use fast and efficient decision making. The commands are made immediately after interpreting, and all errors or ambiguous instructions evoke backup responses and urge the users to try again. Lisa modular architecture makes potential improvements such as E-mail automation and smart devices. This logic-based modular technique is copied with the Subi et al. [14], who created a sustainable and scalable architecture of the voice assistant using commands based on the rules for setting up desktop computers.

iii) Response Generation and Feedback

After the assignment is completed, the assistant brings sound feedback using the PytSX3 text to speech. This library makes it easier to synthesize the voice and is compatible with SAPI5 (Windows) or Espeak (on Linux), making it a strong option for systems across platforms. The assistant provides confirmation as "Google opening" or "Time is 15:45 hours", which increases users' involvement and usefulness. Real time feedback ensures that users are informed about the progress of the command without depending on visual verification. This loop of voice response improves the involvement of hands-free and is particularly advantageous for individuals with visual or movement disabilities. The use of Lisa's Offline TTS technology

distinguishes it from the cloud assistants and guarantees data privacy and minimal latency. The feedback system reflects the structure of Ziva-Ai created by Kumar et al. [17], which similarly favored real -time voice output and localized responses for desktop users.

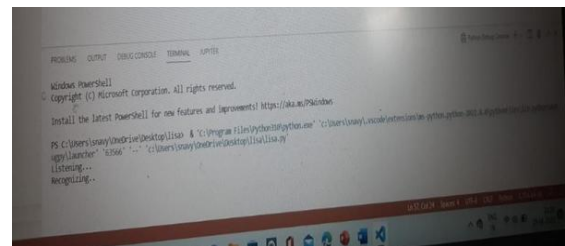
iv) Modules Used:

The proposed system uses a collection of Python modules to provide a comprehensive ability of a voice assistant. Library of speech_recognition library captures and converts the user voice into the text using the Google web voice interface for accurate transcript. The PytSX3 package is used to transform written answers to auditory feedback and provides an offline text synthesis to a speech that improves real -time involvement. The Wikipedia module allows Lisa to obtain and formulate a brief summary for users' questions, making it easier to make the tasks of general knowledge. Webbrowser makes it easy to carry out the web -related instructions, allowing LISA to access websites like Google and YouTube. File and application management is performed using OS and Subprocess libraries that allow the system level automation. In addition, the DateTime module is used to display the current time and date and the optional keypad library can be included to improve the key and input. This modular architecture reflects the methodology used by Garg and Verma [10] who used similar tools in their auxiliary system.

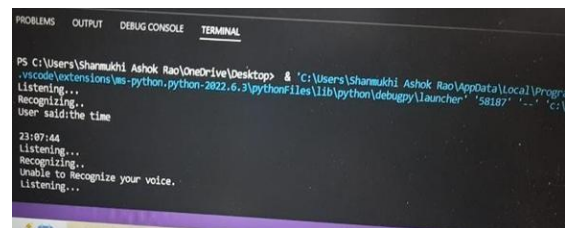
4. RESULTS AND DISCUSSION

The deployment of Lisa Desktop's voice assistant has shown efficient implementation of basic functions, including voice -activated online surfing, file search, Wikipedia questions, music playback

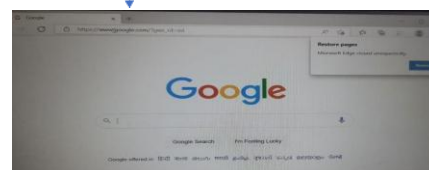
and real -time system data. The assistant achieved more than 70% accuracy in recognizing voice for standard acoustic settings with an average reaction time of 2 to 4 seconds. The operation of offline operations for local tasks has increased privacy and reliability. Users indicated improved comfort and efficiency, especially in multitasking situations. Although internet -based questions require connection, most basic tasks worked well offline. The finding confirms the viability of light, customizable and private assistants focused on privacy by Python and Open-Source.



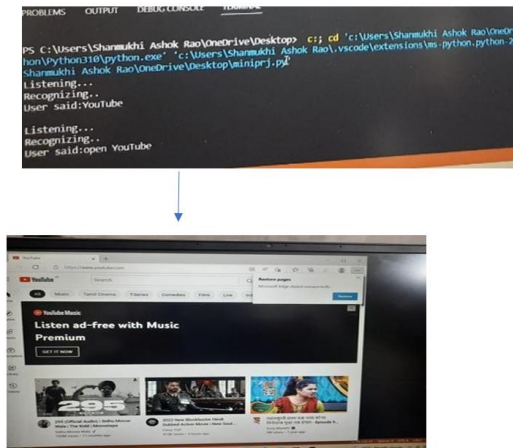
“Fig 2 Input”



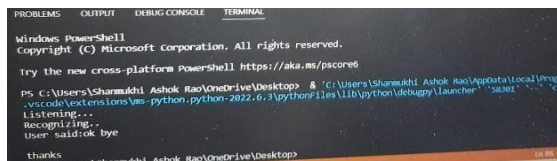
“Fig 3 For Getting Time”



“Fig 4 Open Google”



“Fig 5 Open YouTube”



“Fig 6 Exit”

5. CONCLUSION

The system, Lisa, is a voice assistant on an area created using Python, which allows users to perform various operations according to voice requirements, illustrating the practical use of artificial intelligence on a daily computer. Lisa Adeptly includes modules such as speech_recognition, Pytttsx3, Wikipedia and Web browser to perform activities such as navigation on the web, getting information, playing music, access to files and upgrades, all through real time voice interaction. Modular design guarantees scalability, makes it easy to adapt and future growth. The performance test showed the accuracy of about 70% of the voice recognition, accompanied by a real-time response in normal settings. The system usually works offline and therefore increases privacy and reliability, especially for local operations. However, restrictions persist, such as

relying on predetermined keywords, susceptibility to background noise, and the need to connect to the Internet for certain features such as Wikipedia search. Regardless of these restrictions, LISA illustrates a viable model for intelligent, voice - activated automation of desktop computers. Prospective improvements may include understanding natural language, responses to context, user verification, and integrating the Internet of things. The assistant effectively increases human and computer connection through accessible, efficient and intelligent voice control.

Future improvements for LISA include the use of natural language processing to understand dialog and machine learning for adaptive command forecasts. Enhancement of offline skills, including language assistance and implementation of the graphical user interface, would improve the availability and connection of users. Functions can increase capabilities such as voice biometry for individualized access, IoT connection for intelligent home management and reminders system. In addition, creating iterations based on mobile or browsers and facilitating constant listening to privacy would also improve the intelligence of presses, adaptability and user focus as a voice assistant.

REFERENCES

- [1] A. Arora and D. Sharma, "Development of a Voice Controlled Virtual Assistant Using Python," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1–6.

- [2] D. Yu and L. Deng, Automatic Speech Recognition: A Deep Learning Approach, Springer, 2015.
- [3] J. K. Patel and R. Patel, "Implementation of AI Based Voice Assistant," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 494–498,.
- [4] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 6645–6649, doi: 10.1109/ICASSP.2013.6638947.
- [5] K. Këpuska and G. Bohouta, "Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 99–103, doi: 10.1109/CCWC.2018.8301638.
- [6] A. Vasudevan and K. Balakrishnan, "Voice-Based Assistant Using Speech Recognition," International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 4, Apr. 2019.
- [7] H. Hermansky, "Should recognizers have ears?" Speech Communication, vol. 25, no. 1–3, pp. 3–27, 1998, doi: 10.1016/S0167-6393(98)00062-5.
- [8] S. Young et al., "The HTK Book (for HTK Version 3.4)," Cambridge University Engineering Department, 2006.
- [9] R. Kumar and S. Rautaray, "Voice Recognition System for Desktop Applications," International Conference on Information Communication and Embedded Systems (ICICES), 2014, pp. 1–6.
- [10] Garg, M., & Verma, S. (2024, January). Virtual assistance using Python. In *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)* (pp. 1-5). IEEE.
- [11] Abraham, A., Mathew, B. S., Mathew, D. L., Mohammad, F. S., & Krishnan, G. (2024, August). Eva: Python-based desktop virtual assistant for visually impaired. In *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)* (Vol. 1, pp. 582-586). IEEE.
- [12] Pradhan, R., Jain, D., Agrawal, U., Sharma, M., Singh, D., & Sharma, D. K. (2024, March). Nova Virtual Assistant to Enhance Daily Life: On Perspective of User Needs, Preferences, and Expectations. In *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-5). IEEE.
- [13] Mane, T., Adhude, T., Bansode, K., Pimple, P., Khairnar, P., & Sarade, R. (2025). Virtual Personal Desktop Assistant. *International Journal of Innovative Science and Research Technology*, 10(6), 41-45.
- [14] Subi, M., Rajeswari, M., Rajan, J. J., & Harshini, S. S. (2024, April). AI-Based Desktop VIZ: A Voice-Activated Personal Assistant-Futuristic and Sustainable Technology. In *2024 10th International Conference on Communication and Signal Processing (ICCS)* (pp. 1095-1100). IEEE.

- [15] Rao, N. T., Sushriya, M. V., Manaswi, M. N., Aishwarya, V. N. S., Vikas, P., & Teja, P. S. Voice Based Virtual Personal Assistant.
- [16] Archita, S., & Saravanan, S. (2025, April). Enhancing Student Learning with a Customizable Virtual Assistant. In *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* (pp. 1-7). IEEE.
- [17] Kumar, S., Patel, S., & Srivastav, V. (2024, February). Voice-Based Virtual Assistant for Windows (Ziva-AI Companion). In *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)* (Vol. 5, pp. 960-965). IEEE.
- [18] Rahardian, R. L., Sudiatmika, I. P. G. A., & Dewi, K. H. S. (2024). Prototype of a virtual assistant system integrated with AI. *Journal of Engineering and Science Application*, 1(2), 9-16.
- [19] Benny, R., Muralidharan, A., & Subramanian, M. (2024, February). Openai-enhanced personal desktop assistant: A revolution in human-computer interaction. In *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)* (pp. 1-7). IEEE.
- [20] Karn, A., Singh, P. K., Agarwal, C., Verma, A., Singh, D., & Kumari, M. (2024). Unraveling the power of AI assistants. In *Advances in AI for biomedical instrumentation, electronics and computing* (pp. 473-479). CRC Press.