IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Virtual Bot a Personal Desktop Assistant

Kaustubh Lakade
Department of Electronics
and telecommunication
Vishwakarma Institute of Technology

Khushi Bangad
Department of Electronics
and telecommunication
Vishwakarma Institute of Technology

Sarang Jad
Department of Electronics
and telecommunication
Vishwakarma Institute of Technology

Vedant Malve Department of Electronics
and telecommunication
Vishwakarma Institute of Technology

Vishwakarma Institute of Technology, Pune 411037, Maharashtra, India

Abstract*— *Voice control is a major growing feature that change the way people can live. The voice assistant is commonly being used in smartphones and laptops. AI-based Voice assistants are the operating systems that can recognize human voice and respond via integrated voices. This voice assistant will gather the audio from the microphone and then convert that into text through the Speech recognition , later it is sent through GTTS (Google text to speech). GTTS engine will convert text into audio file in English language, then that audio is played using play sound package of python programming Language.The main task of a voice assistant is to minimize the use of input devices like keyboard, mouse, touch pens, etc. This will reduce both the hardware cost and space taken by it.

Keywords— Desktop Assistant, Python, Text to Speech, Speech to Text, Voice Recognition ,Voice control, AI-based Voice Assistant, GTTS Engine, Playsound, Python.

I. INTRODUCTION

In the 21st century, human interaction is being replaced by automation very quickly. One of the main reasons for this change is performance. There's a drastic change in technology rather than advancement. In today's world, we train our machine's to do their tasks by themselves or to think like humans using technologies like Machine Learning, Neural Networks, etc. Now in the current era, we can talk to our machines with the help of virtual assistants. There are companies like Google, Apple, Microsoft, etc with virtual assistants like Google Now, Siri, Cortana, etc. which helps their users to control their machine by just giving input in the form of voice.

These types of virtual assistants are very useful for old age, blind & physically challenged people, children, etc. by making sure that the interaction with the machine is not a challenge anymore for people. Even blind people who couldn't see the machine can interact with it using their voice only[1].

The concept of virtual assistants in earlier days is to describe the professionals who provide ancillary services on the web. [1] The job of a voice is defined in three stages; Text to speech; Text to Intention; Intention to action; Voice assistant will be fully developed to improve the current range.[6] Voice assistants are not befuddled with the virtual assistants, which are people, who work casually and can therefore handle all kinds of tasks. Voice Assistants anticipate our every need and it takes action, Thanks to AI based Voice Assistants.

Sending and checking email, performing a Wikipedia search, making and receiving phone calls, streaming music, opening programmes, sending texts, general chat, and providing the most recent news updates are some of the fundamental functions that the majority of virtual assistants support.

AI-based Voice assistants can be useful in many fields such as IT Helpdesk, Home automation, HR related tasks, voice based search etc., and the voice based search is going to be the future for next generation people where users are all most dependent on voice assistants for every needs. In this proposal we have built the AI-based voice assistant where we have used python modules and libraries . This assistant is just a basic version that could perform all the basic tasks mentioned above but current technology is although good in it is still to be merged with IOT for better Enhancement.

AI-based Voice assistants can be useful in many fields such as IT, Helpdesk, Home automation ,HR related tasks, voice based search etc, and the voice based search is going to be the future for next generation people where users are all most dependent on voice assistants for every needs. In this proposal we have built the I-based voice assistant where we have used python modules and libraries. The assistant is just a basic version that could perform all the basic tasks mentioned above but current technology is although good in it is still to be merged with IOT for better enhancement.

The voice assistant we have developed is a desktop-based built using python modules and libraries. This assistant is just a basic version that could perform alt the basic tasks which have been mentioned

above but current technology is although good in it is still to be merged Internet Of Things(IoT) for better enhancements.

With the help of voice-activated virtual assistants, there will be no need to write long codes to perform a task, the system will do so for us. The machine will work in three modes-supervised unsupervised or reinforcement learning depending upon the usage for which the assistant is developed. This is all possible with the help of machine learning.

Machine Learning is just a subset of Artificial Intelligence. This has been one of the most helpful advancements in technology. Before AI we were the ones who were upgrading technology to do a task but now the machine is itself able to counter new tasks and solve it without need to involve the humans to evolve it.

II. LITERATURE REVIEW

Personal desktop assistants have grown in popularity due to their ability to boost user productivity and provide personalised assistance. The use of natural language processing and machine learning algorithms has advanced in recent years, leading to the development of more sophisticated personal assistants that can understand user needs and provide relevant information and recommendations.

Huhns and Singh [1] emphasised the importance of designing personal assistants with user needs and context in mind. They emphasised personal assistants' ability to automate routine tasks, provide access to relevant information, and provide personalised assistance.

Marais and Bharat [2] demonstrated the potential of personal assistants in supporting cooperative and personal web browsing. They developed a desktop assistant that used a combination of browsing history, user interests, and collaborative filtering techniques to provide personalized recommendations for web content. This system was found to be effective in enhancing user experience and facilitating knowledge sharing among users.

Hammel et al. [3] assessed the efficacy of a desktop robotic assistant designed to assist people with disabilities with daily tasks. The system employed robotic arms to perform a variety of tasks, including liquid pouring and object picking. According to the study, the system increased user independence and quality of life.

Swamy et al. [4] created a voice- and gesture-based virtual desktop assistant for people who are physically challenged. To allow users to interact with the assistant using voice and hand gestures, the system used natural language processing and gesture recognition algorithms. This system has the potential to improve accessibility and the user experience for people who are physically challenged.

Bui et al. [5] created a context-aware personal desktop assistant that combined natural language processing, machine learning, and decision-making algorithms to provide personalised recommendations and assistance to users in more efficiently managing their work tasks. The system was discovered to be effective in improving user productivity and the overall user experience.

Singh et al. [6] proposed a voice-activated desktop assistant written in Python. The system uses the Google Text-to-Speech engine for speech synthesis and provides voice commands for opening applications and searching the web. The authors assessed the system's accuracy and performance, achieving a recognition accuracy of 90% and a response time of 3 seconds.

Sutar et al. [7] used the Android platform to create an intelligent voice assistant. The system converts speech to text using Google's speech recognition API and supports a variety of voice commands for controlling the device's features. The authors evaluated the system's performance and discovered that it had an 87% recognition accuracy and a response time of 2 seconds.

Subhash et al. [8] proposed an artificial intelligence-based voice assistant that interprets user queries using natural language processing techniques. The system is linked to Google Assistant and offers personalised responses based on the user's preferences. The authors evaluated the system's performance and found it to be 92.4% accurate.

A. Sudhakar Reddy et al. [9] used artificial intelligence and machine learning techniques to create a virtual assistant. The system recognises speech using the Google Cloud Speech-to-Text API and provides voice commands for controlling home appliances and scheduling tasks. The authors evaluated the system's performance and found it to be 93.4% accurate.

O'Shaughnessy [10] reviewed the current state of the art in automatic speech recognition and synthesis, discussing various techniques and algorithms used in the development of voice-based interaction systems. For robust and efficient voice-based interaction, the author emphasised the importance of integrating multiple modalities.

Overall, these studies show that personal desktop assistants have the potential to improve user productivity, user experience, and knowledge sharing. Such assistants must be designed with user needs, context, and advanced natural language processing and machine learning algorithms in mind. Personal assistants are likely to play an increasingly important role in increasing user productivity and improving the overall user experience as technology advances.

III. METHODOLOGY

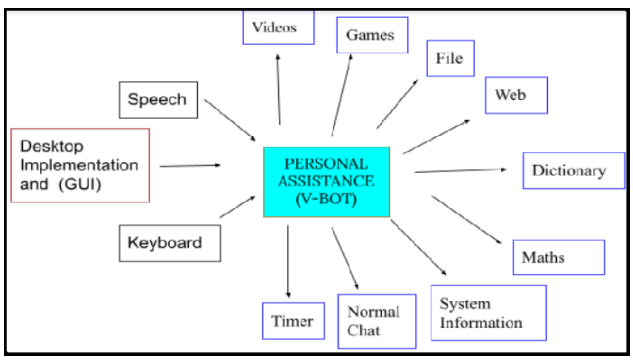

Figure 1:Block Diagram of the Project

The text-to-speech and with the help of Voice User Interface (VUI), which is used to listen to and process audio commands, we have developed a feature called Intelligent Personal Assistant that can carry out background tasks and handle user queries related to their routine needs while working on a desktop. As IoT, smart homes, and linked devices gain popularity, virtual assistants are starting to become a need in daily life. Customers may now connect with technology in a more natural and intuitive way thanks to them. Additionally, virtual assistants can be helpful in a variety of industries, including customer service, education, healthcare, and more. The virtual assistant was built using the Python modules pyttsx3, OS, Tkinter, datetime, webbrowser, wikipedia, PIL, and threading. The time between inputs is determined by the offline text-to-speech conversion module pyttsx3. Operating system-specific functionalities for handling file operations are provided by the OS module. The most popular way to create GUI programmes in Python is by using the Tk GUI toolkit, which has a standard Python interface called Tkinter. Date and time manipulation classes are supported by the datetime module. The webbrowser module enables the system to show users web-based data. The Wikipedia module enables the virtual assistant to respond to questions about Wikipedia. Numerous different image file formats now have support for opening, editing, and saving thanks to the PIL library. A programme can conduct many tasks simultaneously in the same process area thanks to the threading module. The virtual assistant may execute a variety of duties and communicate with users in a more natural and straightforward way thanks to these modules, which complement the virtual assistant's design and implementation.The virtual assistant's design has a user interface that accepts input from users via keyboard or voice commands. Using the pyttsx3 module, the speech input is transformed into text, which is then processed to produce the desired results. A query's execution is decided by one or two keywords that are extracted from the text input. The virtual assistant asks the user to speak again if the keyword does not match any of the coded inquiries. A number of Python modules are used to implement the virtual assistant, including OS for interacting with files, webbrowser for showing users web-based information, wikipedia for handling Wikipedia-related inquiries, and PIL for working with images. The graphical user interface, which makes it easier for users to communicate with the virtual assistant, is created using the Tkinter module. The threading module is also used to conduct many processes simultaneously in the same process area, enhancing the efficiency of the virtual assistant's work. Overall, the virtual assistant's layout makes it simple and intuitive for users to access a variety of tasks, such as file management, online browsing, and information retrieval.Several Python modules had to be integrated in order to change the virtual assistant's background settings. We utilised the Tkinter module to change the colour since it offers a variety of colour possibilities and lets the user choose the colour of their choice. The user may adjust the volume to their desire by using the Pygame module that we used to regulate the volume. The voice rate could be changed using the pyttsx3 module, giving the user control over how quickly the virtual assistant talks.We utilised the ttkthemes module, which offers a variety of theme settings and enables the user to quickly choose between a dark or light theme, to change the theme (from light to dark). The user had the option to choose their favorite voice thanks to the usage of the pyttsx3 module, which also allowed switching from a male to a female voice.A customised virtual assistant that can be tailored to the user's tastes was made possible by the integration of these elements.The Google Maps API was used in the desktop virtual assistant project to provide the maps functionality. With the help of this function, customers may find local businesses like restaurants or tourist sites or ask for directions. We utilised the Python requests module to submit queries to the Google Maps API and obtain the required data in order to build this functionality. Additionally, we used the geopy module to transform human-readable addresses into geographical coordinates that the Google Maps API could use.The virtual assistant analyzes the text input when the user types a question about maps using speech-to-text (STT) and natural language processing (NLP) techniques to extract the pertinent data, such as the location, destination, or kind of site the user is interested in. Using the data it has just retrieved, the virtual assistant then makes a request to the Google Maps API and displays the response in the user interface. The user has the option to interact with the map, zoom in or out, and get more details about the interesting locations.Overall, the maps feature improves the virtual assistant's performance and user experience by giving users a quick and user-friendly method to explore their surroundings.Users of the math function can enter numbers orally or by typing them in to execute mathematical calculations. It evaluates the input as a mathematical expression using the Python 'eval()' function and then returns the outcome. For instance, if the user asks, "What is 2+3?" or "Calculate 2 to the power of 3," the assistant will recognise the question and will then extract the appropriate mathematical expression, evaluate it using 'eval(),' and then deliver the answer. The math function is capable of performing a wide range of mathematical operations, including multiplication, division, exponentiation, and more. It offers customers a practical way to complete rapid calculations without needing to launch a different calculator programme. The News function was introduced to give users access to the most recent news upon request. The virtual assistant responds to the user's voice command, "Give me some news," by opening a web browser, going to a news website, and scraping the most recent headlines to present to the user. Using BeautifulSoup, this functionality asks Python modules to scrape the news website's headlines, and the webbrowser module to launch a browser window. In order to read the whole story, the visitor can then click on the headlines. The purpose of this function is to keep the user abreast of the most recent news without requiring them to actively search through various news sources.

To ensure that the virtual assistant worked as intended, we blended manual and automated testing methods. We performed a range of test scenarios during manual testing to evaluate the performance of the virtual assistant's multiple features, including file manipulation, web surfing, and information retrieval. We also evaluated the accuracy of the text-to-speech and speech-to-text conversions to ensure that the virtual assistant was processing and displaying user input accurately. The virtual assistant was deployed on a local PC by installing all necessary dependencies, including Python, the relevant Python modules, and any additional software tools required for testing and development.

Making sure that the deployment method worked with various operating systems was a difficulty. The virtual assistant's Python foundation made it reasonably simple to verify that it worked with many operating systems. To make sure the virtual assistant operated properly and offered the same functionality on all platforms, we still had to test it on various operating systems.Making sure the virtual assistant could manage a high amount of customer inquiries while maintaining performance and response times was another problem during rollout. Finally, we encountered issues with security and data privacy. We wanted to make sure the virtual assistant was safe and guarded against possible security risks like injection attacks or cross-site scripting attacks because it was processing user input. We overcame this difficulty by establishing safe coding procedures and employing strategies for input validation to thwart fraudulent input.Overall, the virtual assistant's implementation needed careful planning and testing to make sure that it functioned properly and gave consumers a dependable and user-friendly experience, while also addressing any possible difficulties related to compatibility, performance, and security.

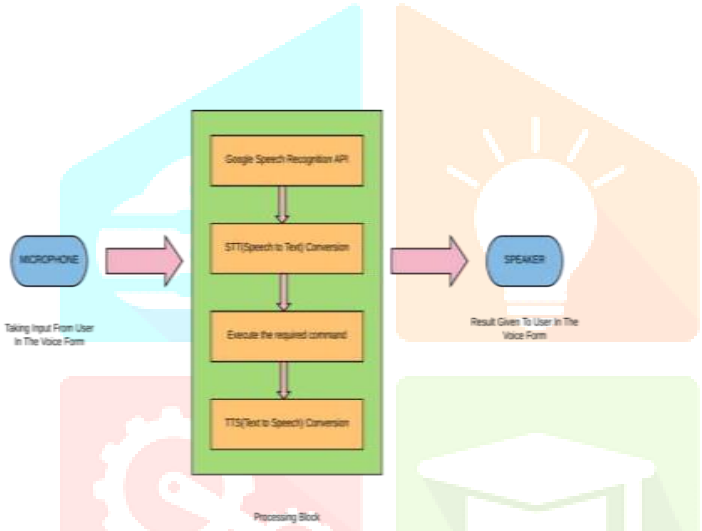

Figure 3:Processing Block of STT To TTS

In order users to engage with the virtual assistant using voice commands, we've created a speech-to-text (STT) to text-to-speech (TTS) processing block in our project. For STT, we utilized the free Python module offline pyttsx3, which was set up using the pip command. When a user talks, the pyttsx3 module turns their speech into text, which the virtual assistant then processes to decide what to do next. The virtual assistant then creates a response in the form of text, which is then translated back to speech using the same pyttsx3 module for TTS after the action has been decided upon.

The Python modules-based creation process for the desktop virtual assistant was effective in accomplishing its goal. The integration of several features, including text-to-speech conversion, online surfing, and date and time manipulation, was made possible by the coupling of the various modules. The user interface was created with simplicity and friendliness in mind. Utilising software tools and test cases, the testing methodology made sure that the virtual assistant performed as expected. The difficulties that arose during the deployment were overcome. Overall, the study sheds light on the practical uses of virtual assistants created with Python modules. This approach may be used to do more study on virtual assistants, especially in desktop settings.

IV. RESULTS & DISCUSSION

The proposed virtual assistant system has a range of functionalities, including variable listening time for user commands and the ability to prompt the user for repeat inputs if necessary. Users can choose between male and female voices, and the current version supports tasks such as creating files on notepad, Word, or Excel, searching on Wikipedia and dictionaries, creating to-do lists, performing math operations, playing games, and opening web browsers. The system also allows for customization of background color, voice and volume settings. In case the system is unable to understand a user query, it will respond with “I couldn’t understand your query” and prompt the user to repeat the query. The final project includes images that demonstrate the virtual assistant in action, completing various tasks.

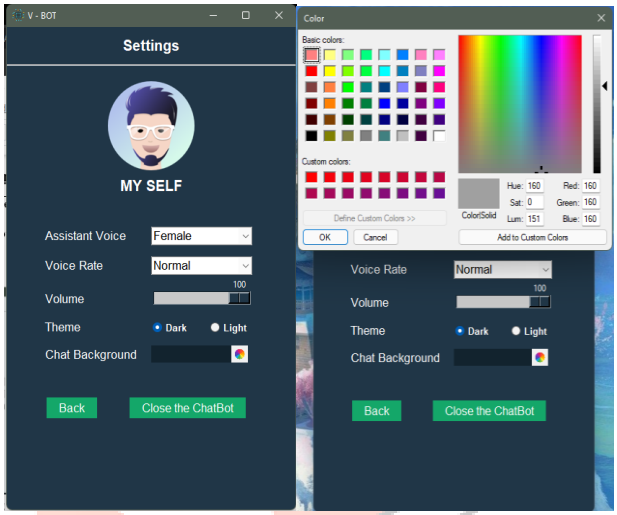

Figure 4:Profile GUI

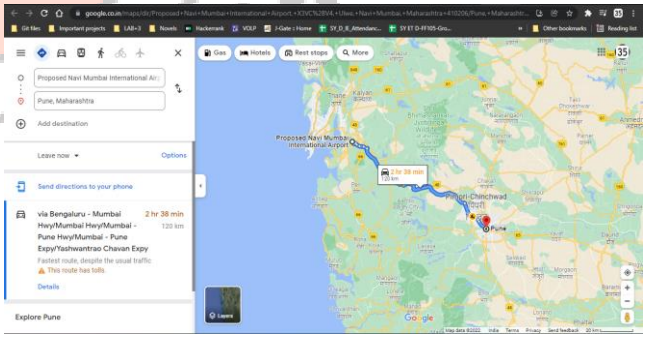

Figure 5:To Find Directions . (Google Map) : Search Directions on Web

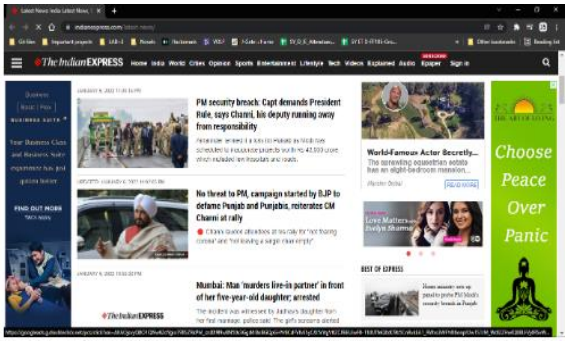

Figure 6:To Give Latest News :

Command Give me Some News : Opens Web Browser For News

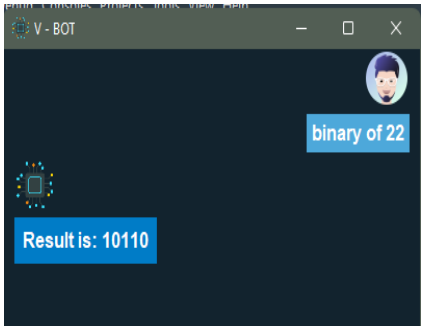

Figure 7: Math Calculations

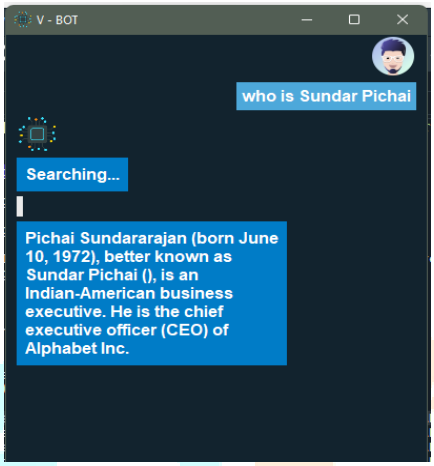

Figure 8:Wikipedia Search

V. FUTURE SCOPE

Given all the potential areas for development and growth, the future scope of this desktop virtual assistant project is substantial. Enhancing the precision and responsiveness of the speech-to-text and text-to-speech modules using more sophisticated machine learning algorithms and methodologies is one potential route for future improvement. Additionally, more sources and information could be added to the maps and news features, giving users access to even more thorough and current information. In order to increase the functionality and usefulness of the virtual assistant, more third-party programmes and services, such social networking websites or online stores, might be incorporated with it. The virtual assistants currently available are fast and responsive, but they still need to be improved. The future of these assistants will incorporate Artificial Intelligence, Machine Learning, Neural Networks, and IoT to achieve new heights. Voice assistants can be used for two developments: quality of dialogue recognition and automation of repetitive tasks, such as opening video conferencing and booking meeting rooms. Broadband allows more complex data processing in powerful data centres, and voice assistants can be used to automate repetitive tasks, such as opening video conferencing and booking meeting rooms.

VI. CONCLUSION

In this paper, we examined a Python-based Voice Activated Personal Assistant. A virtual assistant is software that can carry out tasks given to it by clients and can comprehend spoken or written instructions. In order to match user voice or text input with executable commands, it employs natural language processing (NLP). A virtual assistant is accessible at predetermined hours and is fast to adjust to changing demands. If their workload permits, it can also assist others, such as relatives and coworkers.Currently, this assistant is functional and can be used to open desktop programmes, stream YouTube videos, search Wikipedia, and perform other basic tasks. The existing system can only function as a desktop application at this time. The system will combine machine learning in the forthcoming versions of this assistant, which will provide improved suggestions with IoT to operate the local objects similarly to what Amazon's Alexa and Google home does.

VII. ACKNOWLEDGMENT

We would like to express our sincere gratitude to Prof. Mrunal Shidore for their invaluable guidance, support, and encouragement throughout the course of this DT paper. Their expertise and insights were instrumental in shaping the direction and scope of our work. We are also thankful for their patience and dedication in answering our queries and providing feedback, which significantly improved the quality of our work. We are truly grateful for their unwavering commitment to our academic growth and success.

VIII. REFERENCES

[1] O'Shaughnessy, Douglas. "Interacting with computers by voice: automatic speech recognition and synthesis." *Proceedings of the IEEE* 91.9 (2003): 1272-1305.106699.

[2] Mohadikar, Usha, et al. "Survey on Advance Searching by Virtual Assistant Using Python." *Journal propertypage: www. ijrpr. com ISSN* 2582: 7421.

[3]Paul, Rajdip, and Nirmalya Mukhopadhya. "A Novel Python-based Voice Assistance System for reducing the Hardware Dependency of Modern Age Physical Servers." *IRJET* 8.05 (2021).

[4]Sutar Shekhar, S., et al. "Intelligent voice assistant using android platform." *International Journal of Advance Research in Computer Science and Management Studies* 3.3 (2015).

[5]Subhash, S., et al. "Artificial intelligence-based voice assistant." *2020 Fourth world conference on smart trends in systems, security and sustainability (WorldS4)*. IEEE, 2020.

[6]Manojkumar, Mrs Patil Kavita, et al. "AI-Based Virtual Assistant Using Python: A Systematic Review."

[7]Agarwal, Ayush, et al. "Virtual Voice Assistant using Python." *Journal of Advanced Research in Information Technology, Systems and Management* 6.1 (2022): 5-8.

[8]Pandey, Ankit, et al. "Smart voice based virtual personal assistants with artificial intelligence." *Artificial Computational Research Society* 1.3 (2020).

[9[Dhanraj, Vishal Kumar, and Semal Mahajan. "Research Paper onDesktop Voice Assistant." (2022).

[10[Geetha, V., et al. "The voice enabled personal assistant for Pc using python." *International Journal of Engineering and Advanced Technology* 10 (2021): 162-

165.

[11]Raza, Zeeshan, et al. "VIRTUAL DESKTOP ASSISTANT USING PYTHON."

[12]Appalaraju, Vadaboyina, et al. "Design and Development of Intelligent Voice Personal Assistant using Python." *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. IEEE, 2021.

[13]Biradar, Saurabh, et al. "PERSONAL VIRTUAL VOICE DESKTOP ASSISTANT AND INTELLIGENT DECISION MAKER."

[14]Pal, Sanjima, et al. "Desktop Voice Assistant Using Artificial Intelligence." *American Journal of Electronics & Communication* 3.4 (2023): 12-17.

[15]Janani, N., R. Janet Jessica, and M. S. Vinmathi. "Desktop Assistant."

[16]Sakharkar, Ashutosh, et al. "Python based AI assistant for computer." *Conference on advances in communication and control systems (CAC2S)*. 2021.

[17]Moore, Alan D. *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter*. Packt Publishing Ltd, 2018.

[18]Fitzpatrick, Martin. *Create GUI Applications with Python & Qt5 (PyQt5 Edition): The hands-on guide to making apps with Python*. Martin Fitzpatrick, 2020.

[19]Pant, Tanay, and Tanay Pant. "Understanding and Building an Application with STT and TTS." *Building a Virtual Assistant for Raspberry Pi: The practical guide for constructing a voice-controlled virtual assistant* (2016): 9-19.

[20]Jain, Sanyam, Shivani Sharma, and Ravi Tomar. "Integration of wit API with python coded terminal bot." *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3*. Springer Singapore, 2019.

[21]Jain, Raj Kumar, et al. "Artificial Intelligence Based A Communicative Virtual Voice Assistant Using Python & Visual Code Technology." *World Journal of Research and Review (WJRR)* 13.5 (2021): 23-26.

[22]Yan, Mengting, et al. "Building a chatbot with serverless computing." *Proceedings of the 1st International Workshop on Mashups of Things and APIs*. 2016.