Desktop Voice Assistant

Anushree Nimishe[1], Deeksha Suryawanshi[1], Pallavi Pawar[1], Sheetal Kanathe[1], Vidhi Buwade[1], Prof. Kanchan Narware[2]

[1]Student, Department of Computer Science Engineering, Shri Balaji Institute of Technology and Management, Betul, MP, India
[2]Assistant Professor, Department of Computer Science Engineering, Shri Balaji Institute of Technology and Management, Betul, MP, India

Abstract
In this contemporary age, daily life has been streamlined and made more convenient through the integration of various technologies into home automation systems, including Artificial Intelligence (AI), Cloud Computing, Mobile Computing, voice assistants, as well as image and video processing. This project focuses on utilizing voice input to generate both spoken and text-based output, providing a user-friendly interface to assist individuals in performing tasks on their personal computers. In an era where progress is a central focus of development, our voice assistant is designed to communicate with users in their regional language, with English serving as the default language for accessing files and websites on the system. Through voice commands, the system interprets user instructions and executes operations on the Windows operating system. The application is constructed using AI technology, Python programming, and APIs.

Keywords: Artificial Intelligence, Voice Assistant, Desktop Assistant, Python, Machine Learning.

I. INTRODUCTION

In recent times, automation has surged, facilitating human-machine interaction primarily through voice assistants. Notably, the emergence of new advancements in Artificial Intelligence (AI) has introduced voice assistants like Alexa by Amazon and Siri by Apple. These AI developments enable various speech recognition tasks such as text-to-text, speech-to-text, and speech-to-speech through APIs. Voice assistants excel in performing diverse functions such as sending emails, managing to-do lists, accessing websites or apps, and sending texts solely through verbal commands. A prominent example of desktop voice assistant is Cortana, an intelligent personal assistant and voice recognition software for Windows PC. The adoption of such intelligent automation systems is steadily rising, with voice research taking precedence over text search. This hands-free operation eliminates the need for manual input. Our voice assistant, tailored specifically for the Windows operating system using Python, incorporates a graphical representation through animated presentations. It is engineered for enhanced efficiency and improved human-machine interaction.

II. LITERATURE REVIEW

Each company-developer of the intelligent assistant applies his own specific methods and approaches for development, which In turn affects the final product. One assistant can synthesize speech more qualitatively, another can more accurately and without additional explanations and corrections perform tasks, others are able to perform a narrower range of tasks, but most accurately and as the user wants[1].

The genesis of speech recognition traces back to Bell Laboratories in 1952, with the creation of the first system named Audrey. In the realm of computer science, AI research is defined as the study of intelligent agents. A

voice assistant is a software program that can perform tasks or provide some kind services for an individual based on the human verbal commands i.e. by using human voice commands and the voice assistant will respond via synthesized voice[2].

The history of voice assistants is characterized by multiple waves of significant innovations. Voice assistants for dictation, search, and voice commands have become standard features on smartphones and wearable devices. This study is based on an extensive literature review aiming to provide comprehensive knowledge (theory and concepts) about voice control, virtual assistants, their various applications, and more. As we examine numerous intelligent programs currently available with natural language processing capabilities, we find a multitude of examples integrated into everyday life, serving various functions[3].

III.METHODOLOGY

Fig1. Working of assistant

i. **Python -** Python stands out as an object-oriented, high-level programming language renowned for its integrated dynamic semantics, particularly in web and app development. Its appeal lies in its suitability for Rapid Application Development, owing to features like dynamic typing and dynamic binding. Python's simplicity contributes to its accessibility, as it employs a syntax that prioritizes readability, making it easy to learn. Developers find Python code more legible and translatable compared to other languages, which ultimately reduces the costs associated with program maintenance and development. This accessibility fosters collaborative work among teams, minimizing language and experience barriers.

ii. **NLP (Natural Language Processing) -** Natural Language Processing (NLP) is a facet of artificial intelligence (AI) that encompasses the capability of computer programs to comprehend human language in both its spoken and written forms, commonly known as natural language.

iii. **API (Application Program Interface**) - An application programming interface (API) consists of a set of established guidelines facilitating communication between various applications. It serves as a mediator layer, managing data exchanges between systems. This enables companies to expose their application data and functionality to external third-party developers, business partners, and internal departments within their organizations.

iv. **Pyttsx3** - Pyttsx3 is a Python library used for text-to-speech conversion. Distinguishing itself from other libraries, it operates offline and is compatible with both Python 2 and 3. To obtain a reference to a pyttsx3 Engine instance, an application calls the pyttsx3.init() factory function. It's an effortlessly user-friendly tool that converts text input into speech. The pyttsx3 module offers support for two voices: female and male, provided by "sapi5" for Windows. Additionally, it supports three text-to-speech (TTS) engines.

Algorithm- "Speech Driven Command Interpreter"

1. **Initialize Text-to-Speech Engine and Speech Recognizer:**

- Import necessary libraries: `speech_recognition` for capturing audio input and `pyttsx3` for text-to-speech conversion.
- Initialize the text-to-speech engine (`engine`) using `pyttsx3.init()`.
- Initialize the speech recognizer (`recognizer`) using `sr.Recognizer()`.

2. **Define Command Processing Function:**

- Define a function `process_command(command)` to interpret and execute user commands.
- Inside this function, add logic to handle different commands. In the provided example, it handles the command "open browser" by opening the browser and provides feedback to the user.

3. **Main Loop:**

- Enter an infinite loop to continuously listen for user input.
- Within the loop:
- Use a `with` statement to capture audio from the user's microphone.
- Print a message indicating that the program is listening.
- Use the `recognizer.listen()` method to capture the audio input.
- Try to recognize the speech using `recognizer.recognize_google(audio)`.
- If successful, print the recognized command and pass it to the `process_command()` function for further processing.
- Handle exceptions like `sr.UnknownValueError` and `sr.RequestError` for cases where speech recognition fails or there's an issue with the request.

4. **Process User Commands:**

- Inside the `process_command()` function:
- Check the command against predefined actions or use NLP techniques for more advanced interpretation.
- Execute the corresponding action based on the command.
- Provide feedback to the user using text-to-speech conversion.

5. **Continuous Listening:**

- The program continuously listens for user input until terminated by the user.

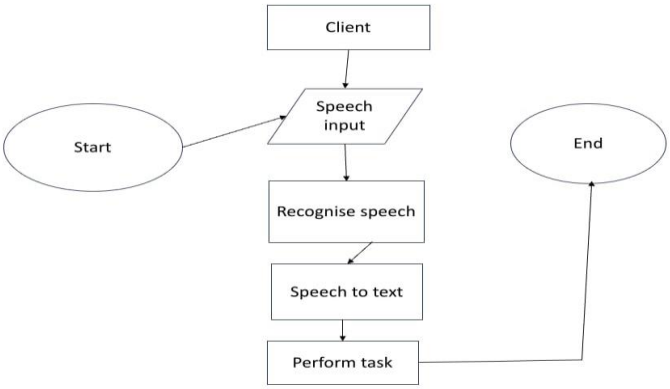

Fig2. DFD For Desktop Voice Assistant

IV.OUTCOMES

A time-saving solution, virtual assistants efficiently interpret instructions and execute tasks delegated by users. Utilizing natural language processing (NLP), they match user's voice or text input with actionable commands, enabling control of machines such as laptops or computers. The swift and effective performance of virtual assistants renders them an optimal choice for time management across various activities. They promptly respond to voice commands or textual inquiries, eradicating the necessity for manual intervention or navigation. This streamlined approach proves particularly beneficial for repetitive or time-intensive tasks, enabling users to allocate their attention to other pressing obligations. Available round the clock, virtual assistants offer reliability and readiness for assistance. They adapt swiftly to evolving requirements, offering emergency aid as necessary. Moreover, virtual assistants accommodate multiple users, including family members or colleagues, contingent upon their permissions and workload. This adaptability extends their utility beyond individual users, facilitating assistance to others as well.

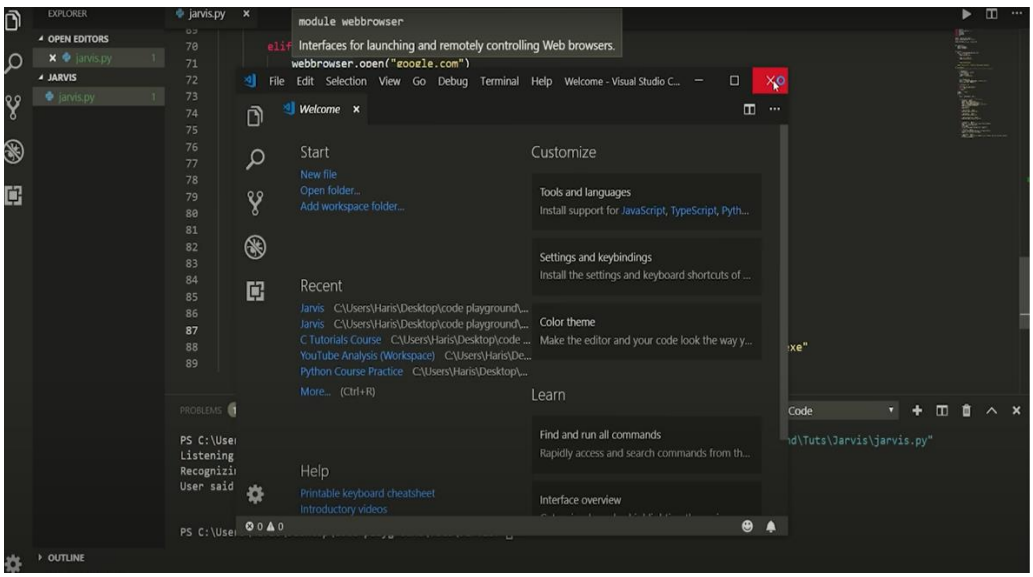

Fig3. Output 1- Opening VS Code

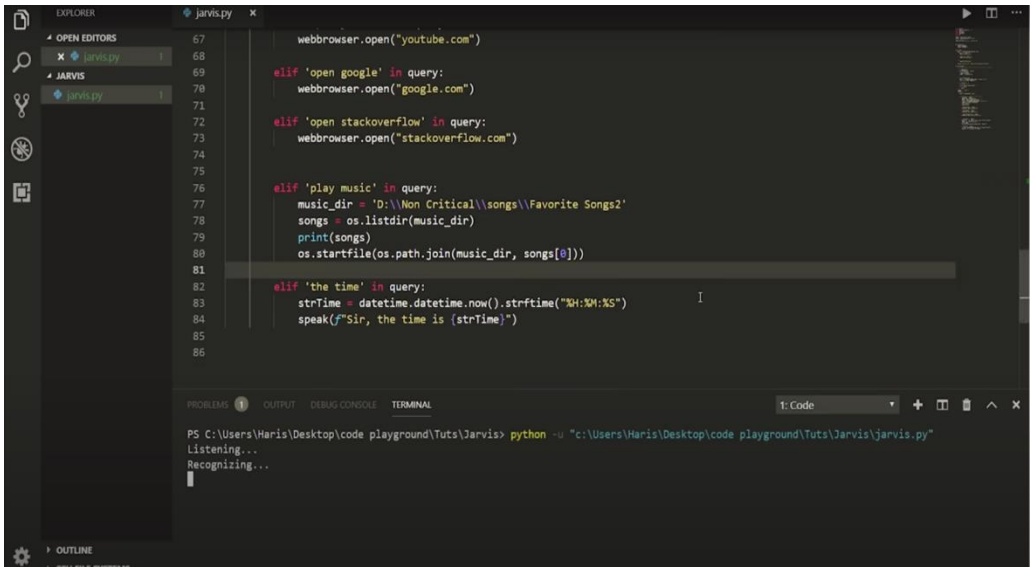

Fig 4. Output 2- Telling time.

V. FUTURE SCOPE

Based on our survey findings, we propose the development of an application tailored to meet the diverse needs of users. The primary motivation behind users' interest in utilizing a voice assistant is to simplify their daily lives. By incorporating the following features, we can enhance user satisfaction:

a. Supporting various languages and accents to cater to a wider user base.
b. Ensuring portability across different environments for seamless usability.
c. Introducing voice authentication technology to bolster security measures.
d. Implementing a chatbot with a robust corpus to enhance conversational capabilities.
e. Enhancing dialogue flow by incorporating neural networks for smoother interactions.
f. Utilizing frameworks like Flask or Django to deploy the application on the web.
g. Leveraging cloud platforms such as Amazon EC2 or Heroku for scalable deployment.
h. Integrating NLP functionalities such as entity recognition and topic modeling to enrich user interactions.

VI. CONCLUSION

In summary, this paper offers an in-depth exploration of the design and implementation of a desktop voice assistant using the Python programming language. By understanding natural language commands, executing tasks, and providing relevant responses, voice assistants hold promise in enhancing productivity and convenience.

Personal desktop voice assistants have the potential to revolutionize how we interact with other devices, simplifying access to information and improving productivity across various tasks.

With this technology, we stand poised to reach new heights. The capabilities of digital assistants surpass our current achievements, presenting opportunities for further advancement. While existing virtual assistants are already quick and responsive, there is still much progress to be made in this field.

REFERENCES

1. Deepak Shende, Riya Umahiya, Monika Raghorte, Aishwarya Bhisikar, Anup Bhange “AI Based Voice Assistant Using Python” From KDK College of Engineering, Nagpur, India. Journal of Emerging Technologies and Innovative Research (JETIR), Volume 6, Issue 2, 2019.

2. Lalit Kumar “Desktop Voice Assistant Using Natural Language Processing (NLP)” From Maharaja Agrasen Institute of Technology, Delhi, India. International Journal for Modern Trends in Science and Technology (IJMTST), Volume 6, Issue 12, 2020(a).

3. Subhash, S., Srivatsa, P. N., Siddesh, S., Ullas, A., & Santhosh, B “Artificial Intelligence-based Voice Assistant” From Dayananda Sagar College of Engineering, Bengaluru, India. Institute of Electrical and Electronics Engineers (IEEE), 2020(b).

4. Mehdi Mekni “An Artificial Intelligence Based Virtual Assistant” From University of New Haven, New Haven, Connecticut, USA. Journal of Software Engineering and Applications (JSEA), 2021.

5. Shivam Singh Sikarwar “AI Based Voice Assistant” From MITS Gwalior, India. International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), Volume 4, Issue 5, 2022.

6. Shubham Thorbole, Anuradha Pandit, Gayatri Raut, Tejas Sirsat “AI Based Desktop Voice Assistant” From SKNCOE, SPPU, Pune, India. Journal of Technologies and Innovative Research (JETIR), Volume 10, Issue 5, 2023(a).

7. Sakshi R Jain, Prof Feon Jason “Personal Desktop Voice Assistant” From Jain University, Bengaluru, India. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 12, Issue 3, 2023(b).