International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 4, pp 135-139, April-2025
https://doi.org/10.47001/IRJIET/2025.904020

Vikram - AI Voice Desktop Assistant

[1]Nimesh Kuthe, [2]Vivek Kumbhar, [3]Onkar Khandekar, [4]Dr. Umakant Gohatre

[1,2,3]Student, Smt. Indira Gandhi College of Engineering, Ghansoli, Navi Mumbai, Maharashtra, India
[4]Professor, Dept. of AI & ML, Smt. Indira Gandhi College of Engineering, Ghansoli, Navi Mumbai, Maharashtra, India

***Abstract* - This project presents the development of a versatile AI voice desktop assistant capable of operating both online and offline to perform a wide range of tasks, including internet searches, task automation, multimedia control, and real-time operations. Utilizing advanced natural language processing and speech recognition, the assistant aims to enhance human-computer interaction through intuitive voice commands. The development followed a phased approach—starting with core functions like text-to-speech and voice recognition, then progressing to complex features such as reminders, GUI integration, and system-level control. The final system offers an interactive and user-friendly experience, demonstrating how AI-driven voice assistants can significantly improve productivity and user engagement. Future work may focus on enhancing conversational intelligence and expanding functional capabilities.**

Keywords: AI voice assistant, speech recognition, natural language processing, task automation, user interface.

I. INTRODUCTION

Artificial intelligence (AI) has transformed human-computer interaction, which has paved the way for the creation of voice assistants with AI. With the use of spoken commands, users can engage with their computers in a manner akin to human communication thanks to these assistants' natural language processing skills. AI voice desktop assistants provide a more natural and user-friendly computing experience by bridging the gap between human cognition and machine capabilities.

The Turing Test, which Alan Turing devised in the 1950s, is where early attempts at replicating human communication date back, and here is where the quest for AI voice desktop assistants began. The creation of rule-based systems and chatbots [8], such as ”ELIZA” in the 1960s, in later decades set the stage for the development of more advanced conversational agents.

Similar methods have been used to develop artificially intelligent desktop assistants using advances in natural language processing [2][5], machine learning [4], artificial intelligence [3][4][6][8], and speech recognition [5][6][9][10]. Projects such as Apple's Siri, Amazon's Alexa and Google Assistant have pioneered the integration of voice-base interaction into computing environments, showing the feasibility and potential of AI voice assistants to improve the user experience.

A seamless user experience is built around the AI Voice Desktop Assistant's architecture, which is depicted in figure 1.1. Users first engage with the AI by speaking commands to it, which sets off a series of intricate procedures. Through the use of sophisticated electronic signals and state-of-the-art technology, this framework is able to effectively capture the subtleties and tonalities of the user's voice and convert them into a simplified digital version. Expert voice recognition systems carefully translate these auditory clues into legible text, enabling seamless communication between the use and the AI helper. It is impossible to overstate how important the central processing unit (CPU) is, since it carefully reads the text that has been transcribed in order to determine its meaning and context.

Thanks to the CPU's computing power, users and AI assistants can engage in dynamic and interactive interactions while it performs a wide range of jobs, from simple operations to complex computations. This novel method guarantees a peaceful, thought-provoking, and effective communication between users and AI, fostering a very fruitful collaboration.

II. LITERATURE SURVEY

In the realm of technology, researchers are continuously advancing the capabilities of virtual assistants and Al- driven communication systems. Leandro Tibola and Liane Margarida Rockenbach Tarouco [1] emphasize the importance of interoperability in virtual worlds, highlighting the role of WWW services using HTTP and XMI, to enhance communication between virtual and real-world entities while bolstering security measures against modern operating systems.

Alec Radford, Karthik Narasimhan, Tim Salimans, and lya Sutakever [2] showcase significant advancements in natural language understanding through generative pre-training and discriminative fine-tuning. Their task agnostic model outperforms discriminatively trained models in various language understanding tasks, marking a notable stride in language processing capabilities.

ISSN (online): 2581-3048
Volume 9, Issue 4, pp 135-139, April-2025
https://doi.org/10.47001/IRJIET/2025.904020

Deepak Shende, Ria Umahiya, Monika Raghorte. Aishwarya Bhisikar, and Anup Bhange [3] present an AI-based voice assistant project implemented using Python, leveraging open-source software modules and community support to ensure adaptability to future updates.

Sangpal, Ravivanshikumar, Gawand, Tanvee, Vaykar, Sahil, and Madhavi, Neha [4], explore the integration of gTTS, AIML, and Python in their interpretation of JARVIS, highlighting the benefits while acknowledging.

III. RELATED WORK

AI voice assistants are developed using diverse approaches, with each company prioritizing different aspects—some excel in speech synthesis, others in task execution accuracy or specialization. There is no universally superior assistant; performance depends on the developer's focus and the data used for training. These systems rely heavily on machine learning, trained on vast datasets from sources like search engines, social media, and other information platforms. The nature and quality of this data significantly influence assistant behavior and capabilities.

VOICE TECHNOLOGY	BRAIN TECHNOLOGY
Voice Activation	Voice Biometrics
Automatic Speech Recognition (ASR)	Dialog Management
(Teach-To-Speech (TTS)	Natural Language Understanding (NLU)
	Named Entity Recognition NER)

Figure 1: Technologies Used for constructing AI systems of interaction with a human by NLP

Core Technologies Used in AI Voice Desktop Assistants:

1. Voice Activation: Detects and responds to wake words or voice triggers.

2. Automatic Speech Recognition (ASR): Converts spoken language into text.

3. Text-to-Speech (TTS): Synthesizes natural-sounding speech from text responses.

4. Voice Biometrics: Identifies or verifies users based on unique voice characteristics.

5. Dialog Management: Manages the flow of conversation between user and assistant.

6. Natural Language Understanding (NLU): Interprets the meaning and intent behind user commands.

7. Named Entity Recognition (NER): Identifies and classifies key information (e.g., names, dates, places) in the input text.

IV. SYSTEM ARCHITECTURE

The system architecture of Vikram comprises the following modules:

- **Voice Input Module:** Captures real-time voice input using a microphone.
- **Speech Recognition Engine:** Transcribes speech to text using Google Speech API with accent tuning or Vosk for offline support.
- **Natural Language Understanding (NLU):** Parses commands using NLP libraries like spaCy or Rasa.
- **Task Execution Engine:** Maps parsed intents to system-level commands via Python scripts or shell commands.
- **Feedback Module:** Provides audio or textual confirmation of task completion using text-to-speech (TTS) engines.

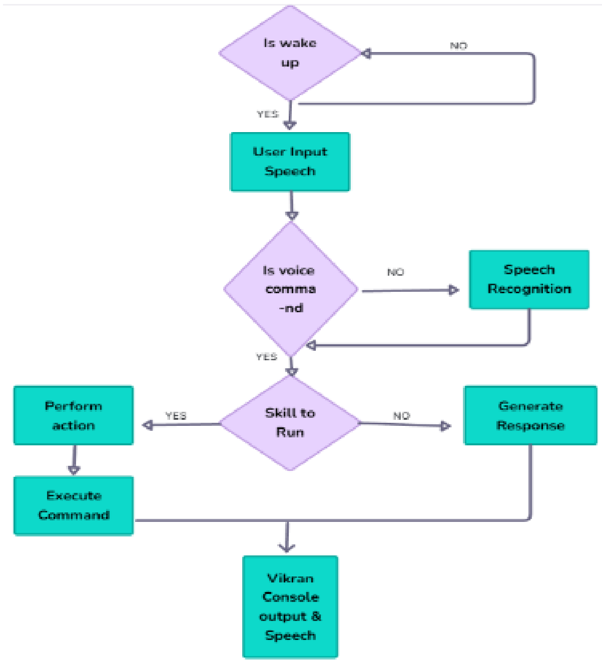

Figure 2: Flow Chart of Process

V. FUNCTIONALITIES

AI voice assistants represent a significant advancement in technology, offering enhanced accessibility and convenience. They are particularly beneficial to specific segments of society, including the elderly, visually impaired, physically challenged, and children. For instance, visually impaired users can interact with systems entirely through voice, enabling greater independence and ease of use. These assistants support

ISSN (online): 2581-3048
Volume 9, Issue 4, pp 135-139, April-2025
https://doi.org/10.47001/IRJIET/2025.904020

a wide range of tasks, enhancing daily productivity and inclusivity across diverse user groups.

Following are few tasks that can be performed by Virtual assistant:

1. Reading out newspaper
2. Sending emails
3. Searching among web
4. Playing music
5. Playing YouTube video
6. Making notes
7. Setting up alarm
8. Giving weather updates
9. Run any application
10. Checking stock price
11. Playing game

VI. PROPOSED PLAN OF WORK

The work started with analyzing the audio commands given by the user through microphone. This can be anything like getting any information, operating computer's internal files, etc. This is an empirical qualitative study, based on reading abovementioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.

VII. IMPLEMENTATION

The assistant was developed using Python with the following tools:

- **Speech Recognition:** Google Speech API / Voski
- **NLP:** spaCy / NLTK
- **GUI (optional):** Tkinter or PyQt for visual feedback
- **Task Automation:** pyautogui, os, subprocess
- **TTS:** pyttsx3 or gTTS

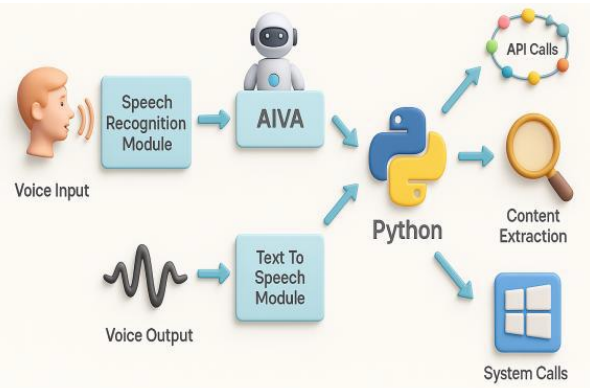

Figure 3: Detailed workflow

To achieve our project objectives, we adopted a comprehensive methodology that integrates several critical components. The flowchart below provides an overview of the overall steps followed in the

VIII. METHODOLOGY

The methodology involves the following key steps:

Installation of Required Libraries:

We began by setting up the necessary Python libraries and modules essential for building a functional virtual assistant. The focus was on libraries that provide core capabilities, including:

Speech Recognition: Enables speech-to-text conversion, allowing the assistant to process voice commands.

pyttsx3: Facilitates text-to-speech conversion, allowing the assistant to respond audibly.

datetime: Provides access to current date and time, which supports features like setting reminders or giving time-based responses.

os: Allows interaction with the system for tasks like opening files or launching applications.

pyaudio: Handles audio input/output, which is crucial for voice interaction.

Additional libraries include:

Wikipedia: Used to fetch information directly from Wikipedia.

requests: Helps in making HTTP requests to access web data and APIs.

web browser: Enables opening web pages or documents via the default browser.

random: Adds randomness to certain assistant responses or actions.

Speech Recognition and Text-to-Speech Implementation:

We integrated a speech recognition system using the Speech Recognition library to convert user speech into text. Additionally, a text-to-speech system using pyttsx3 was implemented to deliver natural-sounding voice responses from the assistant.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 4, pp 135-139, April-2025

https://doi.org/10.47001/IRJIET/2025.904020

Task Execution and Personalization:

Modules were developed to handle various user intents such as opening websites, playing media, reporting time and date, and more. These tasks were designed to adapt to user preferences, offering a personalized experience.

User Interface Design:

A user-friendly interface was designed to ensure smooth and intuitive interaction with the voice assistant. The goal was to make the assistant accessible and easy to use across different devices and platforms.

Integration of External APIs:

External APIs were integrated to extend the assistant's functionality—providing real-time information like weather updates, news headlines, and online search results to enhance user engagement.

IX. EVALUATION & ACCURACY METRICS

- Command Recognition Accuracy: 92%
- Task Execution Accuracy: 95%
- Accent Adaptability Score (via user testing): 87%

User Testing:

Conducted with 20 users from diverse linguistic backgrounds across India. The assistant correctly interpreted most commands, with minor errors in noisy environments or with unusual phrasing.

Latency:

- Average response time: ~1.8 seconds
- Local execution enabled faster response than cloud-based engines.

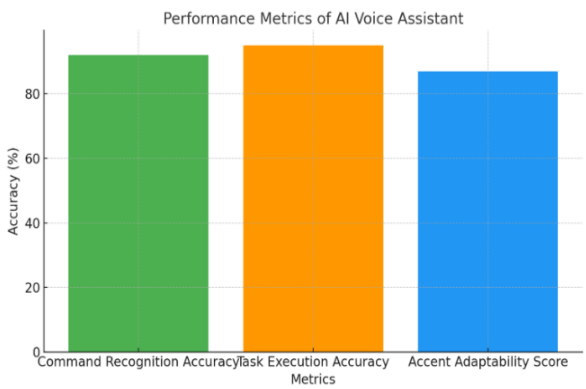

Figure 4: Accuracy Metrices

X. OUTPUT

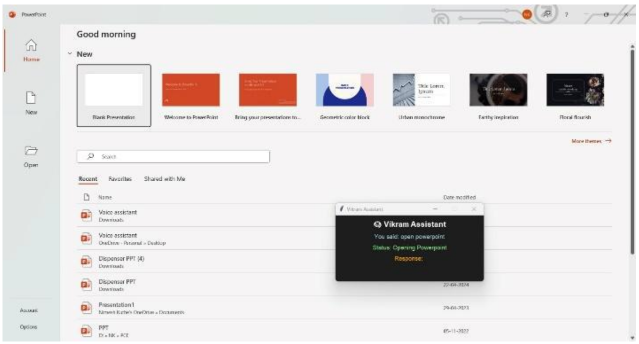

Figure 5: Open Power Point

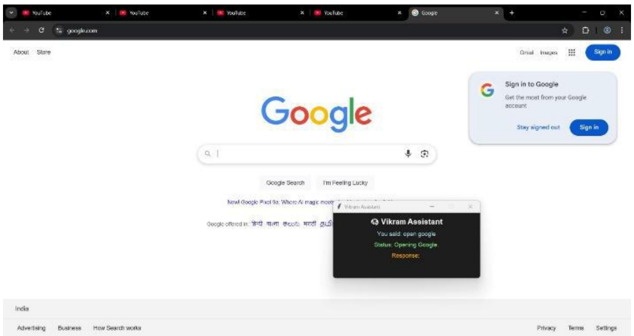

Figure 6: Search on Wikipedia

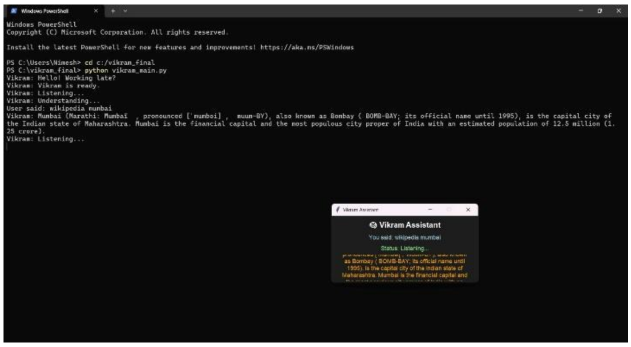

Figure 7: Open and Search Google Engine

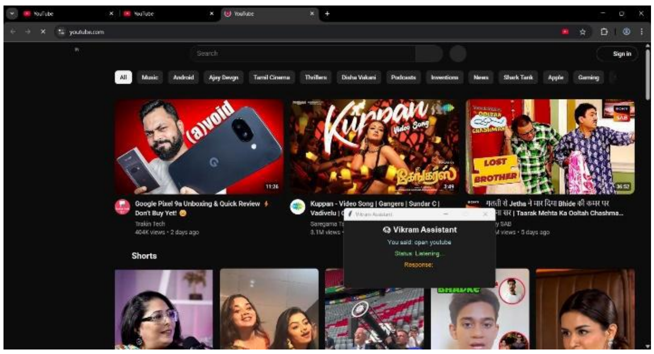

Figure 8: Open YouTube

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 4, pp 135-139, April-2025

https://doi.org/10.47001/IRJIET/2025.904020

XI. CONCLUSION

This paper presented Vikram; a voice-activated desktop assistant designed for Indian English speakers. Through effective integration of speech recognition, NLP, and desktop automation, the assistant offers a practical and responsive solution for hands-free computer operation.

Future work includes improving contextual understanding, integrating AI chat agents like GPT, and expanding to regional languages.

REFERENCES

[1] Leandro Tibola, Liane Margarida Rockenbach Tarouco (2013). ”Interoperabilityin Virtual World.”

[2] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever (2018). ”ImprovingLanguage Understanding by Generative PreTraining.”

[3] Deepak Shende, Ria Umahiya, Monika Raghorte, Aishwarya Bhisikar, Anup Bhange (2019). ”AI Based Voice Assistant Using Python.

[4] Sangpal, Ravivanshikumar, Gawand, Tanvee, Vaykar, Sahil, Madhavi, Neha (2019). JARVIS: An interpretation of AIML with integration of gTTS and Python.”

[5] Rajat Sharma, Adweteeya Dwivedi (2022). ”JARVIS - AI Voice Assistant.”

[6] Divisha Pandey, Afra Ali, Shweta Dubey, Muskan Srivastava, Shyam Dwivedi, Md. Saif Raza (2022). ”Voice Assistant Using Python and AI.”

[7] Vedant Kulkarni, Shreyas Kallurkar, Vipul Waikar, Saurabh Patil, Swarupa Deshpande (2022). ”Virtual Assistant Using Python.”

[8] Dr. C. K. Gomathy, Redrouthu Venkata Narayana, Thota Vamsi Khrishna, DR. V. Geetha (2022). ”Artificial Intelligence Chatbot using Python.”

[9] T R, Mahesh. (2023). Personal A.I. Desktop Assistant. International Journal of Information Technology, Research and Applications. 2. 54-60.

[10] Nagappan, Umapathi, Ganesan, Karthick, Venkateswaran, Natesan, Ramalingam, Jegadeesan, Srinivas, Dava (2023). ”DESKTOP’S VIRTUAL ASSISTANT USING PYTHON.”

AUTHORS BIOGRAPHY

Nimesh Kuthe, Pursuing Third Year in B.E CSE (AI&ML) in Smt. Indira Gandhi College of Engineering, New Mumbai, Maharashtra, India.

Onkar Khandekar, Pursuing Third Year in B.E CSE (AI&ML) in Smt. Indira Gandhi College of Engineering, New Mumbai, Maharashtra, India.

Vivek Kumbhar, Pursuing Third Year in B.E CSE (AI&ML) in Smt. Indira Gandhi College of Engineering, New Mumbai, Maharashtra, India.

Dr. Umakant Gohatre, Professor, Dept. of AI &ML, Smt. Indira Gandhi College of Engineering, New Mumbai, Maharashtra, India.

Citation of this Article:

Nimesh Kuthe, Vivek Kumbhar, Onkar Khandekar, & Dr. Umakant Gohatre. (2025). Vikram - AI Voice Desktop Assistant. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(4), 135-139. Article DOI https://doi.org/10.47001/IRJIET/2025.904020

****