Journal of Electronics and Informatics (ISSN: 2582-3825)
www.irojournals.com/iroei/

Kiki – A Virtual AI Desktop Aider

Nisarga P[1], Sunitha R[2], Pavithra H C[3], Yashashwini Pai K[4]

[1,4]Student, Department of Artificial Intelligence & Machine Learning, BNM Institute of Technology, Bangalore

[2]Associate Professor, Department of Artificial Intelligence & Machine Learning, BNM Institute of Technology, Bangalore

[3]Assistant Professor, Department of Artificial Intelligence & Machine Learning, BNM Institute of Technology, Bangalore

E-mail: [1]Nisargaprakash2002@gmail.com, [2]sunithar@bnmit.in, [3]yaswin2803@gmail.com

Abstract

The research“Kiki” is a voice-controlled virtual desktop assistant designed to enhance user productivity and streamline interaction with desktop environments. Leveraging natural language processing (NLP) techniques, Kiki enables users to perform various tasks through voice commands, ranging from opening applications to searching the web and playing music. The system employs state-of-the-art language models to understand user queries and generate contextually relevant responses, providing a seamless and personalized user experience. Key functionalities include task automation, information retrieval, and web search, allowing users to execute commands efficiently and effortlessly. By combining advanced NLP capabilities with intuitive voice interaction, Kiki represents a significant advancement in desktop assistant technology, promising to reshape the way individuals interact with their computers.

Keywords: Desktop Assistant, Voice Interface, Natural Language Understanding, Task Automation, Conversational AI, Personal Virtual Assistant, Voice Command Recognition.

1. Introduction

The relentless advancement of technology has redefined the landscape of human computer interaction, propelling us towards more intuitive and accessible computing

DOI: https://doi.org/10.36548/jei.2024.2.003
Received: 11.04.2024, received in revised form: 02.05.2024, accepted: 16.05.2024, published: 28.05.2024

experiences. In this era of digital evolution, the integration of voice interaction into computing systems has emerged as a transformative innovation, promising enhanced accessibility, efficiency, and user engagement. This research endeavors to harness the power of voice interaction by integrating it into a desktop aider application developed using Python programming language. The aim is to create a user-friendly device that empowers users to interact with their computers effortlessly, using natural language commands and speech input.

The integration of voice interaction into the desktop aider project is motivated by the desire to address the evolving needs and preferences of modern technology users. By offering an alternative input method through voice commands, the research seeks to enhance accessibility for users with diverse abilities and preferences, fostering inclusivity and usability. Furthermore, voice interaction offers an intuitive and hands-free interface that simplifies the user's interaction with the desktop aide. Users can communicate with the system using everyday language and speech, reducing the learning curve associated with traditional interfaces and enabling more seamless interaction.

Moreover, the integration of voice interaction enhances efficiency and productivity by streamlining the user's workflow. Users can perform tasks, retrieve information, and execute commands quickly and conveniently, without the need for manual input through keyboard or mouse. Through the implementation of voice interaction, the desktop aider research aims to provide personalized assistance tailored to the unique needs and preferences of individual users. By leveraging advancements in natural language processing and artificial intelligence, the aider can understand and respond to user queries effectively, offering a tailored and engaging user experience.

2. Related Work

In the field of virtual assistant technology spans various domains, including assistive technologies for visually impaired and physically handicapped individuals, personalized voice assistants, and desktop virtual assistants. [1] Previous research has explored methods such as text-to-speech (TTS) and speech-to-text (STT) systems to enhance accessibility and independence for visually impaired users, with advancements in AI and deep learning enabling more sophisticated virtual assistants. [2] AIML has been utilized for conversational agents, [3]

while efforts have also focused on regional language support [6] and sign language integration. [9] SVM algorithms have been employed to improve speech recognition accuracy in real-time applications. [7] Customization features enable personalized interactions and services based on user preferences, [8] and deep learning techniques enhance sign language recognition. [10] Voice-activated desktop assistants offer hands-free interaction, [12] [17] and AI-based systems provide intelligent and interactive assistance tailored to individual needs. [18] Integrating voice recognition and face detection capabilities further enhances desktop assistant functionalities. [19] These advancements aim to improve accessibility, productivity, and user experience across diverse user populations.

3. Introduction to Key Components

1. **pyttsx3:** pyttsx3 is a text-to-speech conversion library for Python.

   Functionality: It allows you to synthesize natural-sounding speech from text strings in Python scripts.

   Usage: We use pyttsx3 to provide spoken feedback and responses to users in voice-controlled desktop aider project.

2. **Speech_Recognition:** It is a library used in Python for recognising speech.

   Functionality: It provides easy-to-use functions for capturing audio input from microphones and converting it into text strings.

   Usage: We use speech_recognition to capture and process spoken voice commands from users in voice controlled desktop aider project.

3. **Requests:** requests is an HTTP library for Python.

   Functionality: It allows you to make HTTP requests to web servers and interact with web APIs.

   Usage: We use requests to fetch data from external web APIs or perform web scraping tasks in voice controlled desktop aider project.

4. **Datetime:** datetime is a module in Python's standard library.

   Functionality: It provides classes and functions for working with dates, times, and timestamps.

   Usage: We use datetime to perform tasks such as displaying the current time in voice controlled desktop aider project.

5. **OS:** It is a module in Python's standard library.

   Functionality: It provides functions for interacting with the operating system, such as working with files and directories.

   Usage: We use OS to perform tasks such as opening files or starting external processes in voice-controlled desktop aider project.

6. **Webbrowser:** webbrowser is a module in Python's standard library.

   Functionality: It provides functions for controlling web browsers from Python scripts.

   Usage: We use web browser to open web pages or URLs in the default web browser on the user's system in voice-controlled desktop aider project.

7. **Random:** random is a module in Python's standard library.

   Functionality: It provides functions for generating random numbers and making random selections from sequences.

   Usage: We use random to select random responses or choose random items from lists in voice-controlled desktop aider project.

8. **psutil:** psutil is a cross-platform library for retrieving system information and monitoring system resources.

   Functionality: It provides functions for querying information about system processes and network connections.

Usage: We use psutil to perform system-related tasks such as checking if a specific process is running or monitoring system resource usage in voice-controlled desktop aider project.

9. **gpt4all:** gpt4all is an API for natural language processing (NLP) and text generation.

   Functionality: It provides access to large language models trained on vast amounts of text data, allowing for tasks such as generating human-like text based on user prompts.

   Usage: We use gpt4all to generate responses to user queries or engage in conversational interactions in voice controlled desktop aider project.

4. Proposed Methodology

The code uses the SpeechRecognition library to capture audio input from the user's microphone. sr.Recognizer() is used to create a Recognizer instance. r.listen(source) captures the audio input from the microphone. r.recognize_google(audio) is used to convert the audio input into text using Google's Speech Recognition API. To filter out noise, r.adjust_for_ambient_noise(source, duration=1) is called to adjust the recognizer's sensitivity to ambient noise. This helps in improving the accuracy of speech recognition by reducing background noise interference.

The code analyzes text-based commands using conditional statements (if-elif-else) to determine the user's intent and meaning. Each command is checked against a set of predefined conditions (e.g., 'open reddit', 'play music', 'shutdown', etc.). Based on the detected keywords or phrases in the command, corresponding actions are triggered. For example, if the command contains 'open reddit', the program will open the specified subreddit in a web browser.

Error handling is implemented using try-except blocks to manage potential errors during the execution of the code. In case of an unknown value error during speech recognition (sr.UnknownValueError), the assistant informs the user that their command couldn't be heard and prompts them to repeat it. If an error occurs during HTTP requests (e.g., when fetching a joke from an API), the assistant informs the user and provides a fallback message. If a specific

application (e.g., Notepad) needs to be opened or closed, the code checks if the application is running and handles the situation accordingly.

The trained model used in the code is from the GPT-4-All library, specifically the "orca-mini-3b-gguf2-q4_0.gguf" model. GPT-4-All is a variant of the GPT (Generative Pre-trained Transformer) architecture, designed for natural language processing tasks. The model is fine-tuned on a large corpus of text data to generate human-like responses given a prompt. The dataset used for training the GPT-4-All model likely includes a diverse range of text sources, such as books, articles, websites, and other publicly available text data. The method used in training the GPT-4-All model follows the standard approach for training large language models, which involves pre-training on a large dataset followed by fine-tuning on specific tasks or domains if necessary.

4.1 Features and Capabilities

The virtual assistant offers a range of features and capabilities, including:

Voice Interaction: Users can interact with the virtual assistant using voice commands, enabling hands-free operation and intuitive communication.

Task Execution: The virtual assistant can perform various tasks, such as opening websites, fetching information, playing music, telling jokes, and providing the current time.

Conversational AI: Integration with GPT-4-based conversational AI enhances the virtual assistant's ability to engage in dynamic conversations, respond intelligently to user queries, and maintain context across multiple interactions.

4.2 Code Structure and Functionalities

The virtual assistant's codebase is organized into modular components, each responsible for specific functionalities. The main components include:

Voice Interaction Module: This module handles voice input from the user and processes it for further interpretation. It utilizes the speech_recognition library to capture audio input from the microphone and convert it into text.

Text-to-Speech Synthesis Module: Responsible for converting textual responses into spoken words, the text-to-speech synthesis module utilizes the pyttsx3 library. It enables the virtual assistant to vocally communicate with the user, providing a seamless interaction experience.

Task Execution Module: The task execution module interprets user commands and performs various tasks accordingly. It includes functionalities such as opening websites, fetching information, playing music, and interacting with external APIs. Task execution is facilitated through the integration of relevant libraries and APIs, such as webbrowser, pywhatkit, and requests.

Conversational AI Integration: A key aspect of the implementation is the integration of GPT-4- based conversational AI for handling dynamic conversations and responding intelligently to user queries. The gpt4all library is utilized for interfacing with the GPT-4 model, enabling the virtual assistant to generate contextually relevant responses based on user inputs.

4.3 Algorithm

BEGIN

Initialize the voice-controlled desktop assistant application.

Display "Ready..." to indicate readiness to receive voice commands.

WHILE User continues interaction DO

Prompt the user to speak a command clearly into the microphone. Capture the user's spoken command using the microphone.

Convert the captured audio input into text using speech recognition.

Analyze the text-based command to determine the user's intent and meaning.

Execute the interpreted command by performing the appropriate action.

Generate a response based on the executed command and any relevant information.

Convert the generated response into synthesized speech using text-to-speech module.

Output the synthesized speech to the user to hear the response.

Implement error handling mechanisms to manage any errors that occur during the process.

END WHILE

Terminate the program when the user exits or when finished executing.

END

4.4 Block Diagram

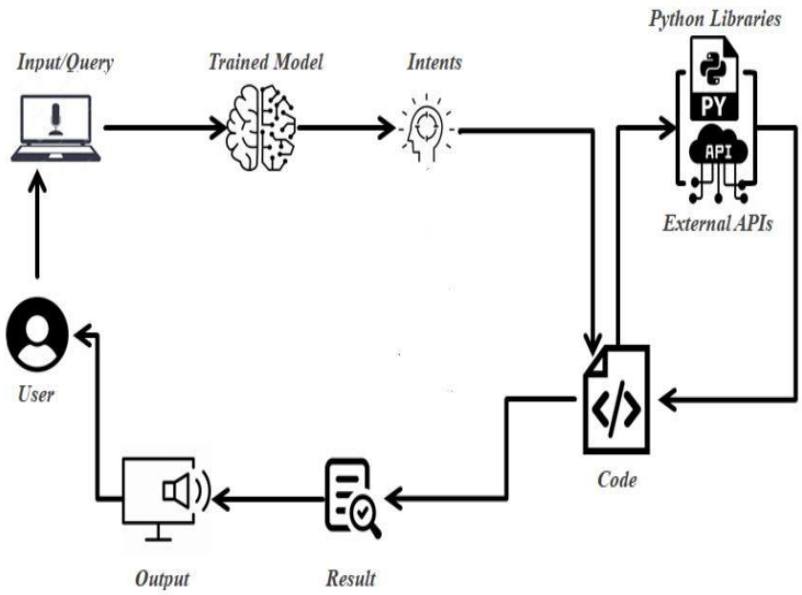

Figure 1. General Block Diagram

4.5 Architectural Design

1. Presentation Layer

- This layer deals with user interaction and presentation of information.
- It includes interfaces such as voice input/output user interfaces.

- Responsible for capturing user commands and presenting responses in a userfriendly manner.

2. Application Layer

- The core logic and business rules of the application reside in this layer.
- It houses modules for various tasks like speech recognition, natural language processing (NLP), command execution, response generation, and error handling.
- Each module is responsible for a specific task in the application's workflow.

3. Infrastructure Layer

- Provides the necessary infrastructure and services to support the application's operation.
- Includes libraries, APIs, and external services used for tasks like speech recognition, text-to-speech conversion, web scraping, system interaction, and communication with external APIs/services.
- Examples include speech recognition libraries, text-to-speech libraries, AI models/APIs for NLP.

5. Result

The AI desktop assistant, crafted by us, stands as a noteworthy advancement in voice-controlled virtual assistants, offering an extensive range of functionalities. Powered by the sophisticated GPT-4 all model, it adeptly responds to voice commands across various domains, providing insightful answers on topics like technology, recipes, stories, and geography. In addition to its informative capabilities, the assistant seamlessly executes practical tasks such as displaying time and date, delivering light-hearted jokes for entertainment, and fulfilling commands to open Gmail, initiate web browsing, or play music on YouTube. Its ability to recognize voice commands and promptly act upon them distinguishes it as a highly responsive and user-friendly interface. While it operates exclusively online, its reliance on voice recognition enhances accessibility and ease of use for users. Overall, the AI desktop assistant

embodies a significant leap forward in voice controlled assistant technology, promising enhanced user experiences and paving the way for further innovations in the field. It allows users to access advanced features in their desktop.

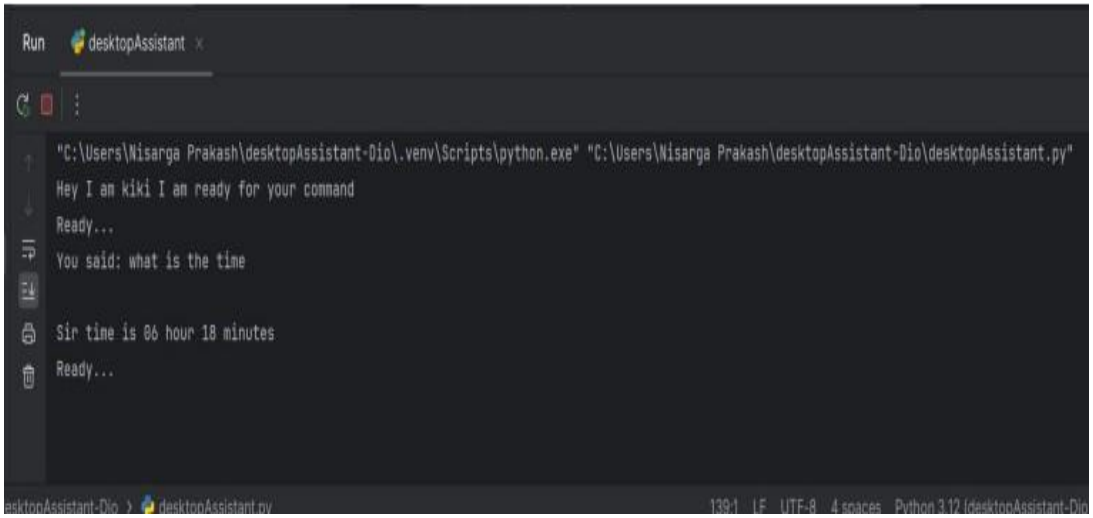

Figure 2. Querying Time

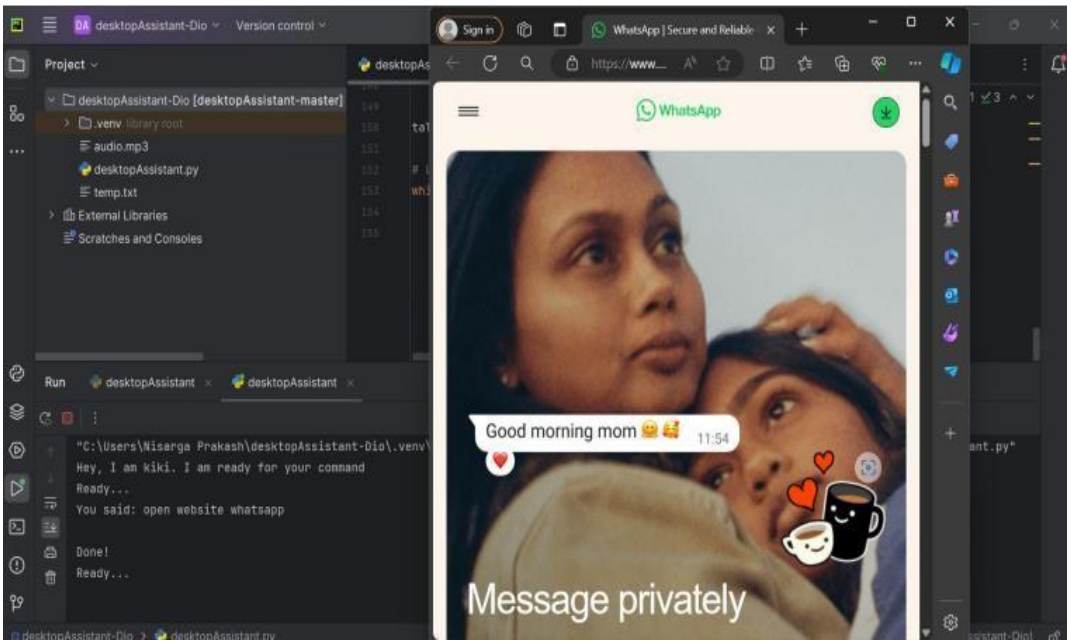

Figure 3. Opening Website

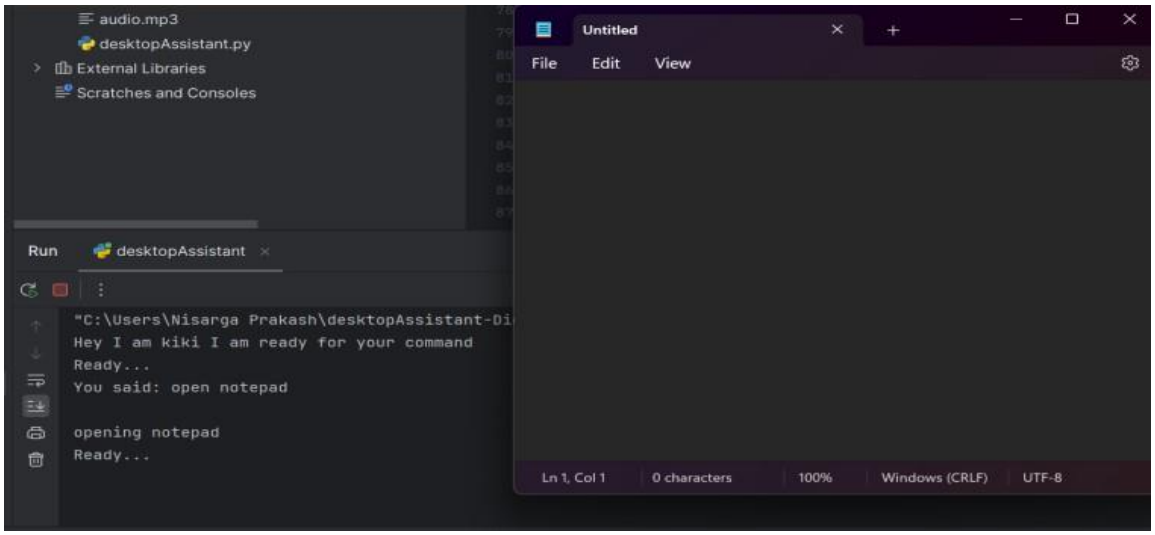

Figure 4. Opening Notepad

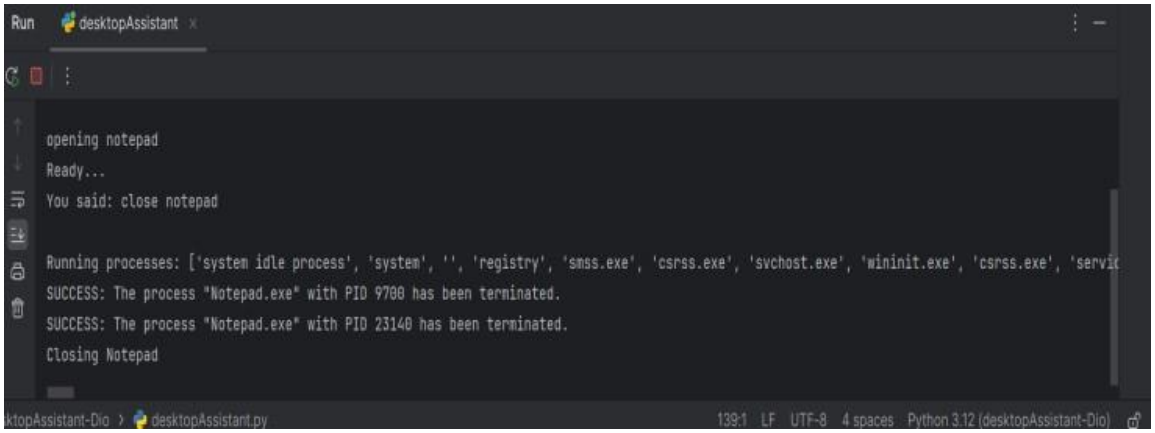

Figure 5. Closing Notepad

Figure 6. Requesting for a Joke

Figure 7. Asking for General Questions

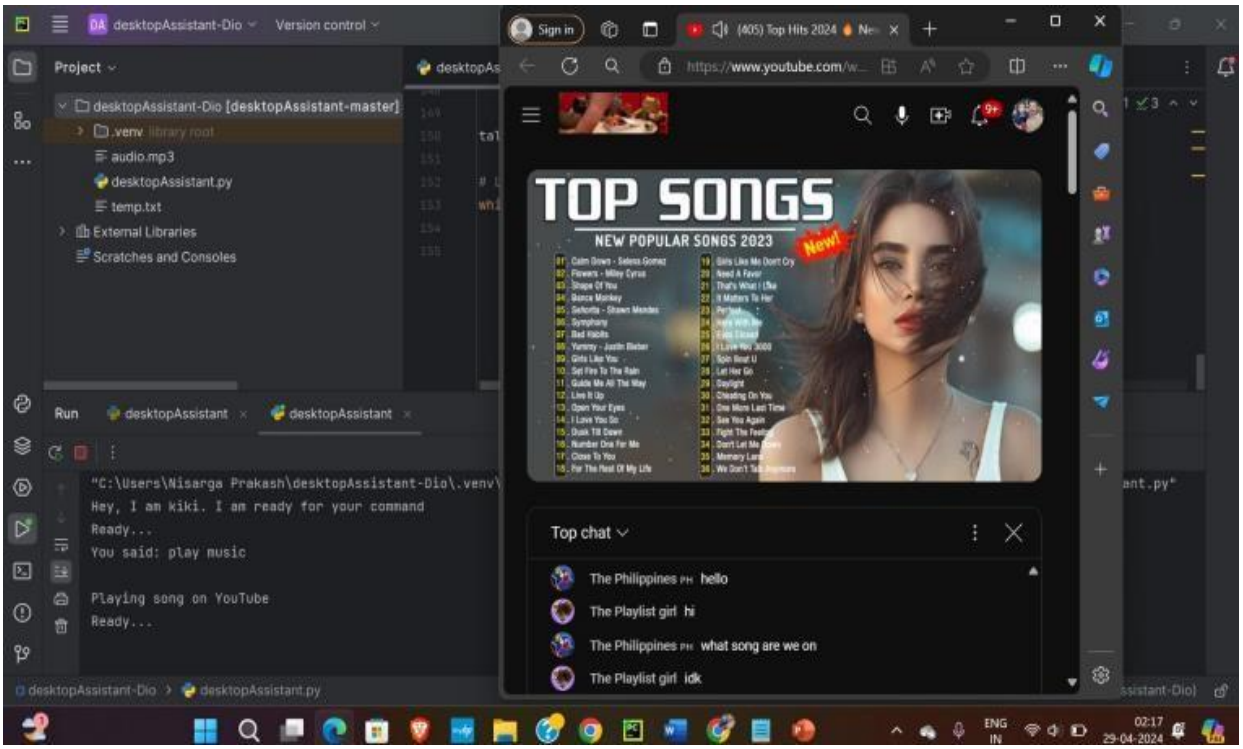

Figure 8. Playing Music

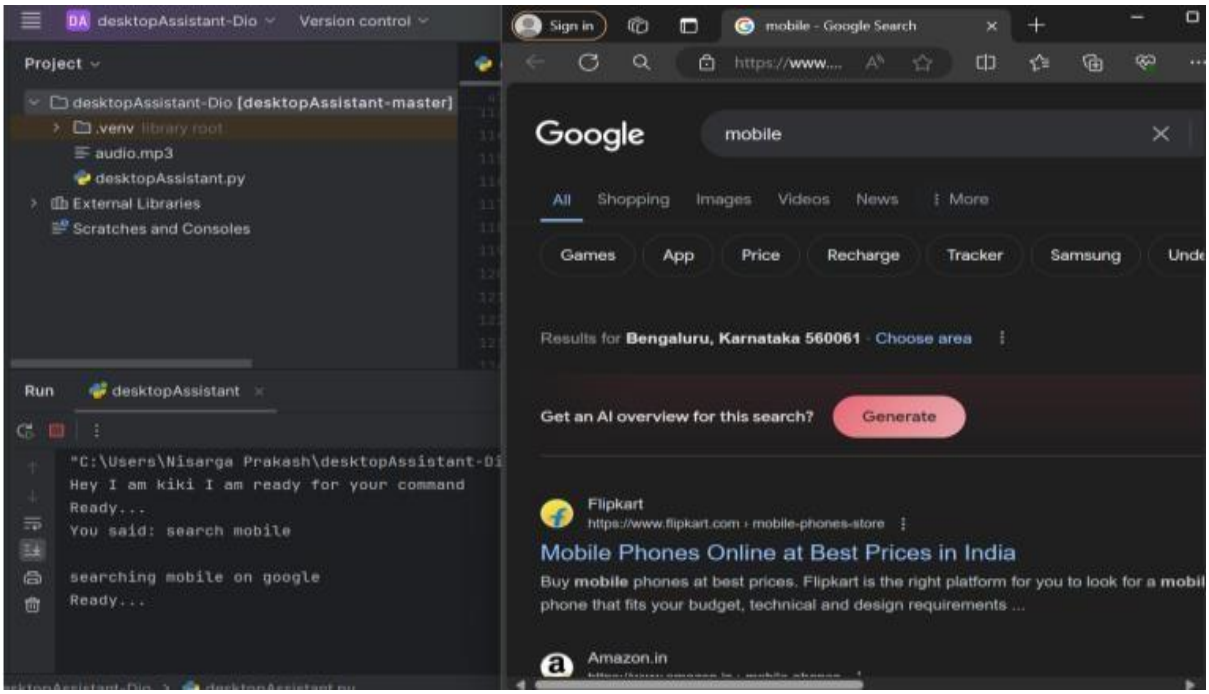

Figure 9. Searching in Google

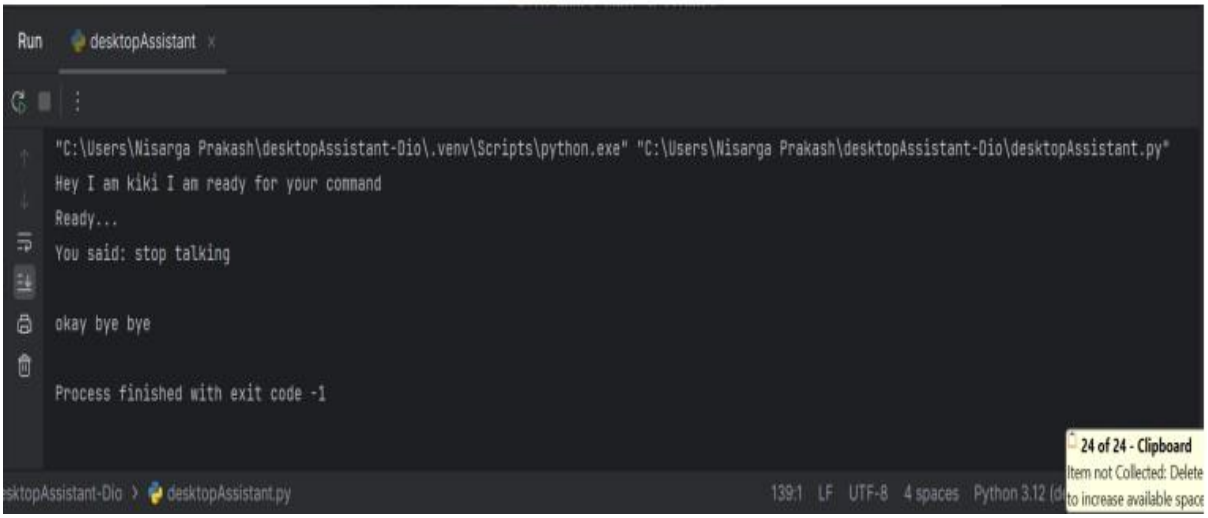

Figure 10. Terminating using Voice Command

6. Comparison with Existing Technology

Table 1. Comparison with the Existing Technology

Paper Name	Technology	Drawbacks	Improvements in Your Project
ABYS(Always By Your Side)	Voice Assistant	Limited capabilities, constrained interactions	Utilizes advanced voice assistant technology (speech recognition and synthesis), integrates with modern AI models like GPT-4
Artificial Intelligence-based Voice Assistant	AI Integration	Reliance on simpler rule-based systems like AIML	Integrates with modern AI models like GPT-4 for more natural and contextually relevant responses
JARVIS: An interpretation of AIML	Application Integration	Lack of integration with specific applications	Integrates with specific applications like Notepad and Brave browser, expanding the range of tasks the assistant can perform
HCI-based Smart Voice Email (Vmail) Application	Entertainment Features	Absence of entertainment features like joke integration or music playback	Includes entertainment features such as joke integration and music playback from YouTube
Voice Email Based On SMTP For Physically Handicapped	AI-powered Conversation	Limited conversational abilities, repetitive responses	Leverages AI-powered conversation using GPT-4, enabling more diverse, contextually relevant, and human-like interactions

7. Conclusion and Future Work

In conclusion, the development of the virtual assistant "Kiki" represents a significant milestone in enhancing user interaction and productivity in the digital age. Through the integration of various Python libraries and APIs, Kiki demonstrates the potential of artificial intelligence and natural language processing in creating intuitive and user-friendly interfaces. The functionalities implemented in Kiki encompass a wide range of tasks, including web browsing, information retrieval, task automation, and natural language conversation. Users can interact with Kiki through speech recognition and receive responses in synthesized speech,

creating a seamless and immersive experience. By leveraging technologies such as speech recognition, text-to-speech synthesis, web browsing, and AI-driven conversation generation, Kiki empowers users to perform tasks efficiently and engage in meaningful interactions. Whether it's fetching information, telling jokes, playing music, or providing assistance with daily tasks, Kiki serves as a reliable and versatile desktop companion. Furthermore, the integration of GPT-4 for conversational AI enables Kiki to generate context-aware responses, making interactions with users more coherent and engaging. This demonstrates the potential of advanced AI models in enhancing virtual assistant capabilities and providing personalized user experiences. Overall, the development of Kiki showcases the power of Python programming and AI technologies in creating intelligent and interactive systems. As technology continues to evolve, Kiki represents a step forward in the evolution of virtual assistants, offering users a glimpse into the future of human-computer interaction. With further refinement and expansion of features, Kiki has the potential to become an indispensable tool for users in various domains, from personal productivity to business assistance. As we continue to explore the possibilities of AI-driven interfaces, Kiki stands as a testament to the ingenuity and innovation driving the advancement of digital assistants.

Moving forward, there are several avenues for enhancing Kiki's capabilities and expanding its utility. These include exploring advanced natural language processing techniques to improve understanding of complex user queries, integrating state-of-the-art machine learning models for generating contextually relevant responses, and developing multi-language support to cater to diverse user populations. Additionally, future work could involve personalization through user profiling, expansion of task automation capabilities, integration with IoT devices for smart home control, and implementation of accessibility features for users with disabilities. Continuous improvement through user feedback, deployment on additional platforms, exploration of novel applications, and ongoing attention to ethical and privacy considerations are also crucial aspects to be addressed in the evolution of Kiki as a versatile and user-centric virtual assistant.

References

[1] M. R. Sultan and M. M. Hoque, "ABYS(Always By Your Side): A Virtual Assistant for Visually Impaired Persons," 2019 22nd International Conference on Computer and

Information Technology (ICCIT), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ICCIT48885.2019.9038603.

[2] S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas and B. Santhosh, "Artificial Intelligence based Voice Assistant," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 593-596, doi: 10.1109/WorldS450073.2020.9210344.

[3] R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, "JARVIS: An interpretation of AIML with integration of gTTS and Python," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 486489, doi:10.1109/ICICICT46008.2019.8993344.

[4] S. Noel, "Human computer interaction(HCI) based Smart Voice Email (Vmail) Application - Assistant for Visually Impaired Users (VIU)," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 895-900, doi: 10.1109/ICSSIT48917.2020.9214139.

[5] S. Kumar, Y. R. and R. Aishwarya, "Voice Email Based On SMTP For Physically Handicapped," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2021, pp. 1323-1326, doi: 10.1109/ICICCS51141.2021.9432206.

[6] M. R. Sultan, M. M. Hoque, F. U. Heeya, I. Ahmed, M. R. Ferdouse and S. M. A. Mubin, "Adrisya Sahayak: A Bangla Virtual Assistant for Visually Impaired," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), DHAKA, Bangladesh, 2021, pp. 597 602,doi:10.1109/ICREST51555.2021.9331080.

[7] Bhosale, R., & Chaudhari, N .Real time enhanced speech recognition technique to operate computer system using SVM.IEEE, May 2017,pp.529-533 R. Bhosale and N. Chaudhari, "Real time enhanced speech recognition technique to operate computer system using SVM,"2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2017, pp. 529-533, doi: 10.1109/RTEICT.2017.8256653.

[8] K. Painchaud and L. Deligiannidis, "Customized Services Using Voice Assistants," 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020, pp. 1060-1065, doi: 10.1109/CSCI51800.2020.00197.

[9] D. Someshwar, D. Bhanushali, V. Chaudhari and S. Nadkarni, "Implementation of Virtual Assistant with Sign Language using Deep Learning and TensorFlow," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 595-600, doi:10.1109/ICIRCA48905.2020.9183179.

[10] Sarathkumar M, Gajendran M, Aravindan B, " Voice Intelligence System (VIS) for disabled", International Journal of Scientific Research in Science and Technology (IJSRST), Print ISSN : 2395- 6011, Online ISSN : 2395-602X, Volume 2, Issue 2, pp.151-153, March-April 2016.Journal URL :https://ijsrst.com/IJSRST162263

[11] K. Sasi Priya, K. Pranathi, C. VinayaKumar, Goutam A Virtual Voice Enabled Assistant for PC using python, e-ISSN: 2395-0056 Volume: 09 Issue: 05 | May 2022 www.irjet.net pISSN: 2395-0072

[12] Alfiya Chougule, Saniya Mulla, Shubham Jadhav, Prof. Parvin Kiniker. ARTIFICIAL INTELLIGENCE DESKTOP VOICE ASSISTANT IN PYTHON R May 2023, Volume 10, Issue 5 www.jetir.org (ISSN-2349-5162)

[13] Mohd Talib Ansari, Ansar Ahmed, Aadil Jiwani, Prajaktee Rane. DESKTOP VIRTUAL-ASSISTANT,2021,pp.3034-3036,DOI Link:https://doi.org/10.22214/ijraset.2022.40785

[14] Bhavya Jain, Arpan Saini, Mr. Kuwar Pratap Singh. F.R.I.D.A.Y – Voice Assistant, e-ISSN: 2395-0056 Volume: 08 Issue: 05 | May 2021 www.irjet.net pISSN: 2395-0072

[15] B. V. Patil and K. Sreelakshmi, "Implementation of Voice Based E-Mail System for Visually Challenged," 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 2022, pp. 1-9, doi: 10.1109/INCOFT55651.2022.10094386.

[16] Indranil Basu, Saumyadeep Bhattacharyya, Arpan Mondal, Babin Maitra, Roshni Joardar, Toyesh Dey, Koushik Pal Voice Activated Desktop Assistant Using Python ,2022 IJCRT | Volume 10, Issue 6 June 2022 | ISSN: 2320-2882

[17] Gowhar Ahmad Dar, Jeby Tom Kurian, Abin K Shaji, Chrisil T Jose, Dr. Anju ratap ARTIFICIAL INTELLIGENCE -BASED VOICE ASSISTANT Volume 7 Issue 3 March 2020 eISSN: 2349-5162

[18] Rabin Joshi, Supriyo Kar, Abenezer Wondimu Bamud and Mahesh T R. Personal A.I.DesktopAssistant.IEEE,2023,pp.5460,DOILink:https://doi.org/10.22214/ijraset. 2023.50132

[19] Raju Shanmugam, Rajesh Kumar Patjoshi, Soumya Ranjan Jena, Vishvaketan Gaur Desktop Assistant Based on Voice Recognition and Face Detection . Vol. 12, Issue 4, April 2023 DOI: 10.17148/IJARCCE.2023.1243