Artificial Intelligence Voice Assistant Using Python, Text to Speech, and GPT

Anirban D. Chakraborty [1]
Department of Information Technology
G H Raisoni College Of Engineering
Nagpur, India
anirban.chk1@gmail.com

Aneesh Bhattacharjee [2]
Department of Information Technology
G H Raisoni College Of Engineering
Nagpur, India
aneeshb35@gmail.com

Dev Mawley [3]
Department of Information Technology
G H Raisoni College Of Engineering
Nagpur, India
devmawley09@gmail.com

Aniket Paulzagde [4]
Department of Information Technology
G H Raisoni College Of Engineering
Nagpur, India
paulzagdeaniket2002@gmail.com

Himani Hingve [4]
Department of Information Technology
G H Raisoni College Of Engineering
Nagpur, India
himanihingve21@gmail.com

Abstract- This research project aims to develop an AI voice assistant using Python, Text-to-Speech (TTS), and OpenAI's GPT-3 language model, along with various modules such as speech_recognition, pyttsx3, webbrowser, wikipedia, and wolframalpha. The project focuses on creating a robust and user-friendly voice assistant that can perform tasks such as answering questions, searching the web, providing information, and scheduling events. The GPT-3 language model is leveraged to enhance the assistant's natural language processing capabilities, allowing it to understand and respond to user queries with greater accuracy and relevance. The TTS module is used to provide high-quality speech synthesis, and pygame is utilized to play back the generated audio. The research also explores various use cases and scenarios in which the AI voice assistant can be deployed, such as in smart homes, healthcare settings, and business environments. Overall, this research project aims to contribute to the development of intelligent virtual assistants that can enhance human productivity and efficiency.

***Keywords* - TTS, GPT-3, Python, Voice Assistant.**

I. INTRODUCTION

Artificial Intelligence (AI) has been a growing field of research in recent years, with its applications ranging from healthcare to finance and beyond. One of the most fascinating and practical applications of AI is the development of voice assistants. Voice assistants, such as Siri, Alexa, and Assistant, have become increasingly popular in our daily lives, as they offer a convenient and efficient way to interact with our devices. Python is one of the most popular programming languages for AI development due to its versatility and ease of use. In combination with various libraries and modules, such as Text-to-Speech and GPT (Generative Pre-trained Transformer), Python can be used to develop powerful and intelligent voice assistants that can perform a wide range of tasks.

The aim of this research project is to develop an AI voice assistant using Python and various modules, including Text-to-Speech and GPT. The voice assistant will be able to interact with users through voice commands and respond accordingly using natural language processing. The project will make use of several key modules, including speech recognition (using the SpeechRecognition library), text-to-speech (using the pyttsx3 library and Text-to-Speech API), web browsing (using the webbrowser library), and search (using the Wikipedia and WolframAlpha APIs). In addition to these modules, the project will also incorporate the OpenAI GPT-3 model, which is a state-of-the-art language model capable of generating human-like text. The GPT-3 model will enable the voice assistant to understand natural language inputs and respond in a conversational manner.

A. Need and necessity:

An Artificial Intelligence Voice assistant using Python, Text-to-Speech, and GPT can be used to develop a conversational agent that can assist users with a variety of tasks. Some of the needs that this type of AI Voice assistant can address include:

- Hands-free operation: Users may need to perform tasks while their hands are occupied, such as cooking or driving. An AI Voice assistant allows them to perform tasks without having to use their hands.
- Accessibility: An AI Voice assistant can help users with disabilities or those with limited mobility to interact with devices and perform tasks.
- Convenience: An AI Voice assistant can simplify tasks by allowing users to communicate in a natural way without the need to navigate through complex menus or interfaces.
- Efficiency: An AI Voice assistant can automate tasks, such as setting reminders, making phone calls, or sending emails, which can save users time and increase productivity.

B. Basic Functions:

An Artificial Intelligence Voice assistant using Python, Text-to-Speech, and GPT can perform a wide range of functions depending on its capabilities and the tasks it has been trained to perform. Some of the common functions that an AI Voice assistant can perform are:

- Natural language processing (NLP): The AI Voice assistant can understand and respond to natural language commands and queries.

- Web search: The AI Voice assistant can search the web for information and provide answers to user queries.

- Personalized recommendations: The AI Voice assistant can provide personalized recommendations based on user preferences and history.

- Task automation: The AI Voice assistant can automate tasks such as setting reminders, making phone calls, sending emails, and scheduling appointments.

- Entertainment: The AI Voice assistant can provide entertainment by playing music, movies, and games.

- Language translation: The AI Voice assistant can translate languages in real-time to facilitate communication between people who speak different languages.

- Personal assistant: The AI Voice assistant can act as a personal assistant, keeping track of appointments, scheduling meetings, and managing to-do lists.

II. OBJECTIVES

The development of an Artificial Intelligence Voice assistant using Python, Text-to-Speech, and GPT has become an important area of research and development in recent years. The objective of this project is to create a conversational agent that can assist users with a wide range of tasks in a natural and intuitive way.

The AI Voice assistant will be able to understand and respond to natural language commands and queries. This means that users will be able to communicate with the AI Voice assistant in a way that is similar to how they would communicate with a human assistant. The system will use Natural Language Processing (NLP) techniques to analyze and understand user inputs, and generate appropriate responses based on its training data and context.

One of the key benefits of an AI Voice assistant is that it can automate tasks, saving users time and increasing productivity. For example, the AI Voice assistant can help users with tasks such as setting reminders, making phone calls, sending emails, and scheduling appointments. [11] By automating these tasks, users can focus on more important tasks and activities.

Another important function of the AI Voice assistant is web search. The system will be able to search the web for information and provide answers to user queries. This can be useful for a wide range of tasks, such as finding recipes, getting directions, or researching a topic.

III. LITERATURE REVIEW

Mohasi et al (2006) say that Text-to-speech technology is a computer-based tool that has the ability to convert written text into speech. It is an important area of research and development that has numerous applications such as assisting visually impaired individuals in accessing digital content, improving the accessibility of public announcements, and enhancing language learning tools. The process of text-to-speech synthesis involves converting any text format, including words, numbers, and abbreviations, into an understandable speech output. There are three main techniques used for text-to-speech synthesis: articulatory speech synthesis, formant speech synthesis, and concatenative speech synthesis. Articulatory speech synthesis is a model that tries to simulate the movements of the vocal tract during speech production. It uses a set of rules and algorithms to generate speech output based on the articulatory movements. This technique requires a deep understanding of the human vocal tract and a large amount of computational power. [1]

As conveyed by *Nguyen et al. (2010)* The field of voice-based assistants has experienced significant growth and innovation due to their high demand in various devices such as smartwatches, fitness bands, speakers, mobile phones, laptops, televisions, and more. Nowadays, most smart devices come equipped with built-in voice assistants. To effectively handle the enormous amounts of data generated, incorporating machine learning and training devices according to their specific uses is crucial. Other technological advancements such as IoT, NLP, and big data access management are equally important. Voice assistants simplify tasks by allowing users to input voice commands, which the assistant can convert to text and execute queries based on keywords. Patrick Nguyen and his team introduced a novel direct modeling approach for speech recognition called Flat Direct Model (FDM) in their paper "Speech Recognition using Flat Models." This approach solves the problem of consistency in spoken sentences and uses template-based features to improve the sentence error rate by 3% absolute over the baseline. Unlike the conventional Markov model, their model is not sequential, making it a groundbreaking development in the field of speech recognition. [2]

Research by *Agrawal et al. "Voice Assistant Using Python"* An International Open Access-revied, Refereed Journal describes how beneficial it is for people if we reduce the use of input devices and rely on voice commands. The research also lists certain tasks that a virtual assistant must follow, and have added them to their proposed approach such as "keep listening to the commands, and the time for listening variable must be adjusted accordingly.", "It should tell the user to repeat itself if it is unable to hear the commands properly.", "Must provide both male and female voices according to the user's convenience.", "Must provide some kind of entertainment to the user to add to its value such as playing some kind of music, searching on Wikipedia, etc.", "Must come in handy for the user's daily works as well. For example sending emails, sending texts, etc." They have achieved these proposals using Python Machine Learning which is a very optimized approach for this [3]. The drawback in their studies is that they have implemented the AI assistant in Linux, and it works only in Linux, which is not a very common Operating System for people who don't belong to a programming background. [3]

Research held by Vivek (2022) gives a very accurate contrast to what is already available in the market and what are the gaps need to be filled. He very rightly describes that using Natural Language Processing (NLP) and Artificial Intelligence(AI) can multiply the productivity of the virtual assistant by tenfolds. He also enlists multiple use cases of the assistant, such as weather detection, scheduling appointment time, trip planning, etc. A unique element in his research that we found was the technique of mapping the words to functions and parameters to create a command that the computer can understand. His model is based primarily on AI to train the new data and make the assistant better with every use, and the use of NLP which is an AI approach to intelligent programming using a natural language such as English. [4] A drawback that we have noticed in his model is that since it is using AI, it will require a lot of datasets and cleaning and preparing that data for public use in assistant will be a costly and hassled task.

Kumar et al. (2019) have discussed the development of a desktop assistant using Python, Text-to-Speech, and Voice Recognition technologies. The researchers utilized Python as a programming language due to its extensive libraries that could be used to implement the commands required for the assistant. The virtual personal assistant developed by Kumar et al. was designed to be used on any Windows explorer, such as Windows 7, 8, and 10. The authors emphasized that speech technology primarily consists of two technologies, namely synthesizers and recognizers. A speech synthesizer takes input and produces an audio stream as output, while a speech recognizer does the opposite by converting an audio stream into a text transcription. Kumar et al. used Text-to-Speech and Voice Recognition technologies to enable their virtual personal assistant to interpret human speech and respond via synthesized voices. In their study, Kumar et al. aimed to create an efficient and user-friendly virtual personal assistant that can perform a wide range of tasks. The assistant was capable of interpreting natural language commands, searching the web, answering questions, and scheduling events. The authors also emphasized the potential applications of such voice assistants in various settings such as smart homes, healthcare settings, and business environments. [5]

The research conducted by Preethi et al. (2022), shows us a more front-end affiliated function of the assistant. Their research consists of a Graphical User Interface (GUI), which is a very good idea since it does not give a very good impression when we always have to see the black window on running the Python code. Other than this, they have multiple unique functionalities in their approach and execution, such as opening a Word document, opening Chrome and surfing through it, and some on the leisure side such as ordering a pizza from dominos and ordering a mobile phone from Amazon. They focus more on a use case-oriented AI Assistant that can handle multiple tasks. Although we fathom the ideas they have implemented, we could see a major loophole in the implementation itself. After researching further about their project and going through the code out of curiosity multiple times, we noticed that they have used pre-defined commands for executing a task. This reduces the accuracy of the AI assistant by a lot since the assistant can only recognize what it has been fed earlier. In this case, if the user has bad pronunciation, or makes a mistake in giving the exact word-to-word command, the assistant won't be able to recognize the command and hence stop functioning. [6]

Another analysis that we went through was by Pooja et al. (2022). They described the necessity of voice assistants in a very adept manner. The necessities in their analysis included, "To enable a highly engaging user experience", "To make application frustration-free", "To personalise your app experience for every user.", "To remove language barriers." Other than highlighting the importance of the virtual assistants, they have showcased several unique features in their execution as well such as a GUI, for which they have used the PyAuto GUI library in Python, WhatsApp messaging feature, for which they have used the pyWhatkit library, using the keyboard module in python, they have given the complete access of the keyboard to the assistant that alone makes a lot of jobs easier for the user. Using all these libraries makes an impact on the user's internet. They have ensured a solution to this using the speed test library that measures the bandwidth of an internet connection. Lastly, they have implemented the OS module in Python that allows the assistant to open any application that the user demands [7]. They have represented their workflow by using easy-to-understand dataflow diagrams which allows a reader with programming knowledge to understand what is happening in the code.

In a research paper by Yuqi Huang, the importance of artificial intelligence voice assistants in contemporary society is analyzed through a method of comparative analysis. Voice assistants have become a significant feature of electronic devices, allowing users to interact with technology through verbal commands. The paper examines the development of intelligent voice assistants and compares the differences between Chinese and foreign voice assistants. The study also explores how voice assistants can assist in people's daily lives, and highlights the importance of understanding the relationship between humans and machines. The research found that people in different countries have varying preferences for using voice assistants but they can significantly help with daily tasks. Ultimately, the paper concludes that voice assistants have a crucial role to play in modern society. [8]

An analysis by BISHNU S. ATAL, MEMBER, IEEE, AND LAWRENCE R. RABINER describes A Pattern Recognition Approach to Voiced—Unvoiced—Silence Classification with Applications to Speech Recognition. In simpler words, it is an analysis conducted to understand how a speech engine or the computer itself recognizes a speech as voiced or unvoiced. Unlike the other research papers we have referenced so far, this research paper is a mathematical representation of speech and its filters. It has distributed human speech to two entities, a high-pass filter, and a low-pass filter. Further, the research tells under which circumstances a computer considers a high-pass filter as a voiced input and vice versa. We have included this research here because it derives the expression for every important statement it makes and also because it gives a good idea about voice scaling which we can use in our project in the future.[9]

In another research by Hill, J., Ford, W.R. and Farreras, which was about real conversations with artificial intelligence, we got a detailed understanding on the differences between Human-Human interaction and Human-AI interaction. It tells us how an AI perceives a conversation vs how a human perceives it. Highlighting the importance of NLP, it emphasizes on the concepts of Instant Messaging (IM) and Computer-Mediated Communication(CMC).[10]

IV. METHODOLOGY

A. *Packages Used:*

1) *Datetime*:
   - The datetime package is a built-in module that allows you to work with dates and times in our code.
   - We can get the current date and time: datetime.datetime.now().
   - Create a new datetime object: datetime.datetime(year, month, day, hour, minute, second).
   - Get the year, month, day, hour, minute, or second from a datetime object: datetime_object.year, datetime_object.month, datetime_object.day, datetime_object.hour, datetime_object.minute, datetime_object.second.
   - Format a datetime object as a string.

2) *Speech recognition*:
   - The ‘speech_recognition’ package is a Python library that provides a simple way to perform speech recognition in Python. This package can be imported into a Python program to recognize speech from various sources, such as an audio file or microphone input.

3) *Pyttsx3*:
   - Interface for converting text to speech.
   - Uses the text-to-speech capabilities of the operating system, making it easy to switch between different speech synthesis engines.
   - Supports different languages, voices, and speech rates.

4) *Webbrowser:*
   - Provides a high-level interface for displaying web-based documents to users. [11]
   - Can be used to open a web page in the user's default browser, or to launch a specific browser.
   - Supports several browser engines, including Chrome, Firefox, and Safari.

5) *Wikipedia:*
   - Provides a Python wrapper for the Wikipedia API, making it easy to access and parse data from Wikipedia.
   - Supports several Wikipedia languages, including English, Spanish, French, and German.
   - Can be used to search for articles, extract summaries, and get links to related articles.

6) *WolframAlpha:*
   - Provides a Python wrapper for the WolframAlpha computational knowledge engine.
   - Can be used to perform calculations, look up facts and definitions, and generate visualizations.
   - Requires an API key to use, which can be obtained from the WolframAlpha website.

7) *OpenAI:*
   - Provides a Python wrapper for accessing the GPT-3 language model from OpenAI.
   - Can be used to generate text, translate languages, and answer questions.
   - Requires an API key to use, which can be obtained from the OpenAI website.

8) *TTS:*
   - Provides a Python library for converting text to speech using the Cloud Text-to-Speech API.
   - Supports several languages and voices, and allows customization of the speech output.
   - Requires authentication and billing to use, which can be set up through the Cloud Console.

9) *Pygame:*
   - Provides a set of Python modules designed for writing games and multimedia applications.
   - Supports graphics, sound, and input devices, such as keyboards and joysticks.
   - Can be used to create interactive applications, simulations, and educational tools.

10) OS:

- Provides a way of using operating system dependent functionality like reading or writing to the file system or starting new processes.
- Includes functions to interact with the file system, such as os.path, which provides a way of working with file paths and names.

11) Dotenv:

- Allows loading environment variables from a .env file.
- Provides a simple way to manage application settings across different environments, such as development, testing, and production.
- Can be used to store sensitive information, such as API keys and passwords, without hard-coding them in the application code.

B. Data Flow Diagram (DFD):

1) DFD (level 0):

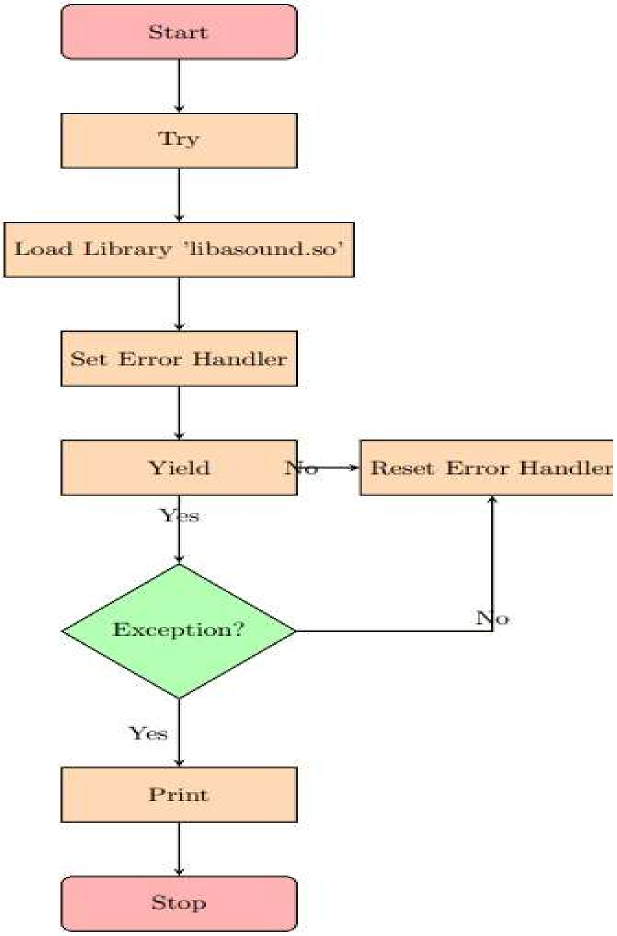


Figure 1: Muting ALSA errors with local speech engine initialization

2) DFD (level 1):

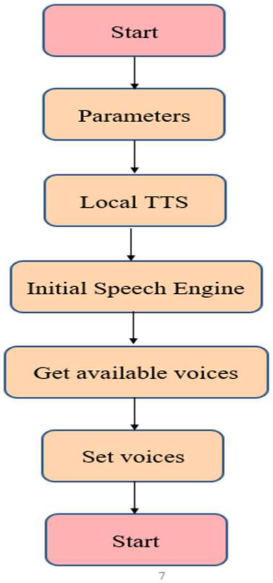


Figure 2: Setting up TTS client

3) DFD (level 2):

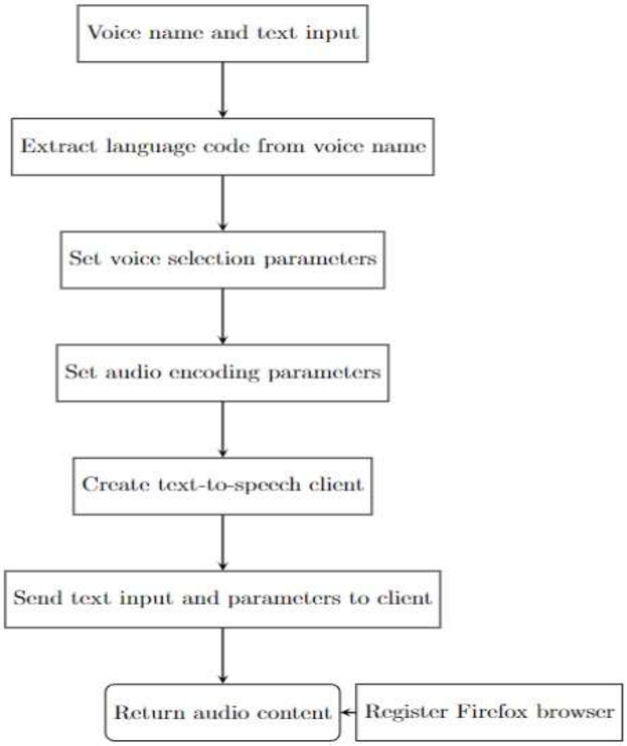

Voice name and text input
Extract language code from voice name
Set voice selection parameters
Set audio encoding parameters
Create text-to-speech client
Send text input and parameters to client
Return audio content
Register Firefox browser

Figure 3: Set text input to be synthesized and configure browser

4) DFD (level 3):

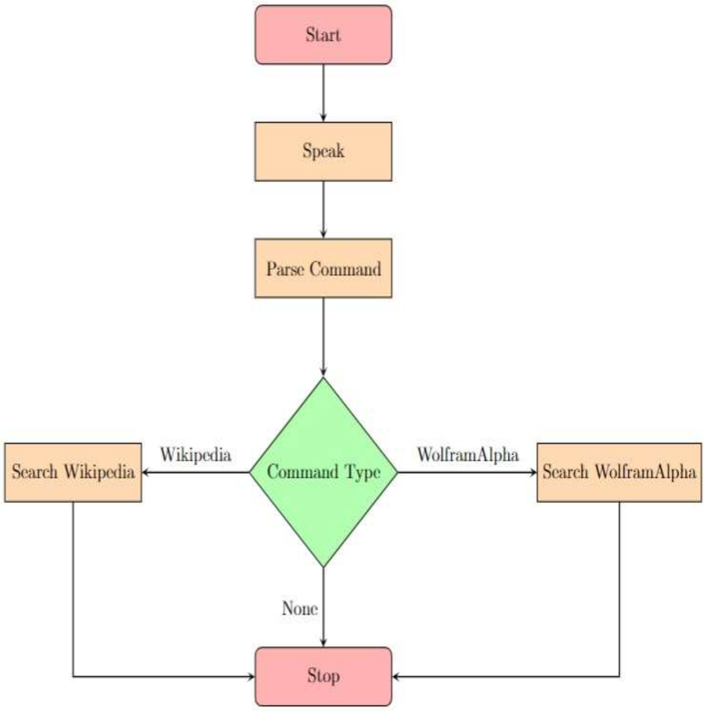


Figure 4: Set up WolframAlpha client and Wikipedia

5) *DFD (level 4):*

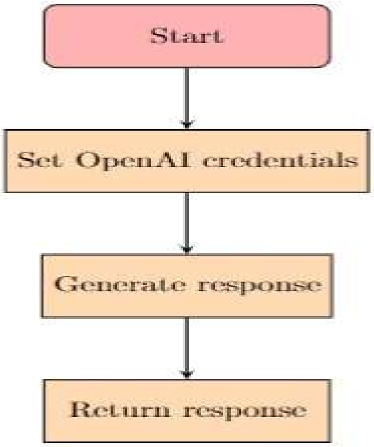


Figure 5: Set up and configure OpenAI API

6) *DFD (level 5):*

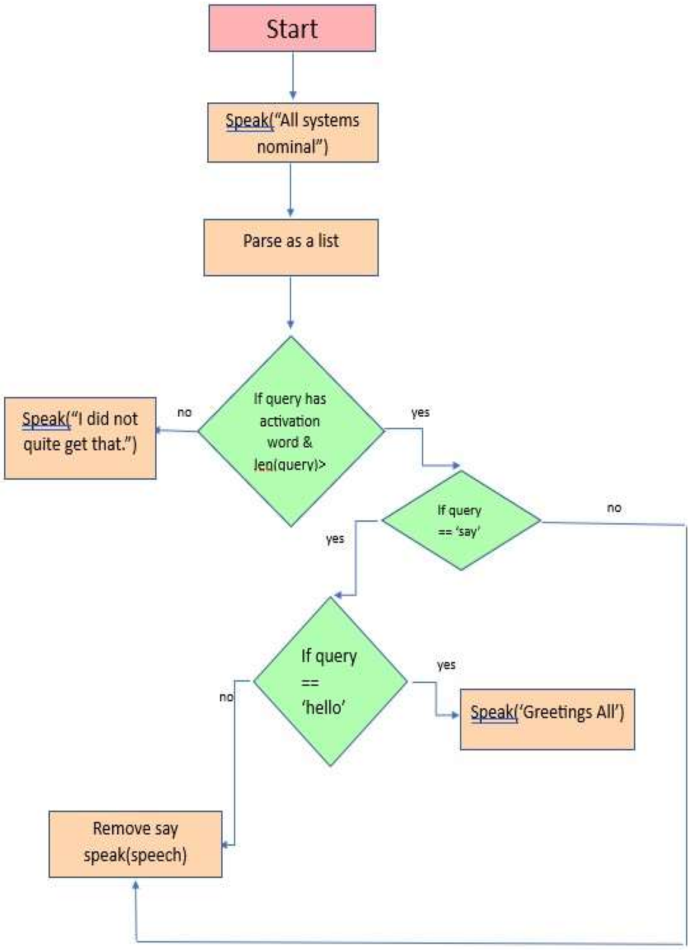


Figure 6: Main loop working and initiation

V. TECHNOLOGIES USED

A.Python:

- We have used python 3.10.10 in this project.
- Python 3.10.10 is a maintenance release in the Python 3.10 series, which was first released in October 2021.
- It includes bug fixes and performance improvements, as well as a few new features.
- One of the major changes in this release is the addition of the zoneinfo module, which provides an improved interface for working with time zones.

B. PyCharm IDE:

- We have used PyCharm version 2022.3.2 for this project.
- PyCharm 2022.3.2 is a powerful integrated development environment (IDE) for Python programming.
- With PyCharm, we were able to easily create and manage Python projects, including setting up virtual environments, installing packages, and running tests.

- PyCharm's code editor is highly customizable, with support for syntax highlighting, code completion, and code navigation.
- Overall, we found PyCharm 2022.3.2 to be an excellent tool for Python development, with a wide range of features and great usability.

VI. FUTURE SCOPE

A. *Improved Natural Language Processing (NLP) and understanding:*

The current implementation uses 's Speech Recognition API for speech-to-text conversion and GPT for text-to-speech conversion. However, there are several other advanced NLP tools such as OpenAI's GPT-3, BERT, and Transformers that can potentially improve the assistant's ability to understand natural language commands and provide more accurate and human-like responses.

B. *More Advanced Dialog Management:*

Currently, the voice assistant can handle single-turn interactions (i.e., a single user input and system response). However, to make the assistant more useful and engaging, it could be extended to handle multi-turn conversations, where the assistant could ask follow-up questions and guide the user toward a specific goal.

C. *Personalization and User Profiling:*

To make the assistant more personalized and tailored to individual users, the system could be extended to create user profiles that store the user's preferences and past interactions with the assistant. This could be used to provide more accurate and relevant responses to the user's requests and improve the overall user experience.

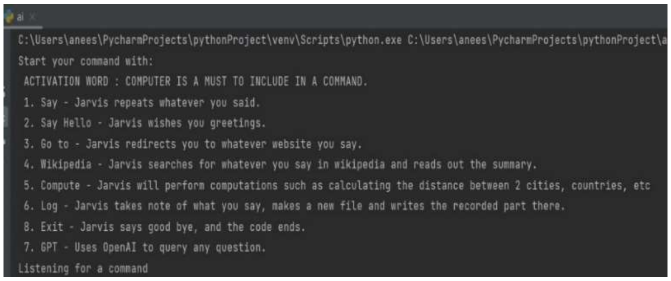
D. *Better Error Handling and Debugging:*

The current implementation does not have robust error handling and debugging capabilities. However, in a real-world scenario, it is crucial to handle errors and exceptions gracefully and provide helpful feedback to the user. Thus, the assistant could be improved with better error handling, debugging, and logging mechanisms to make it more reliable and user-friendly.

E. *Integration with IoT Devices:*

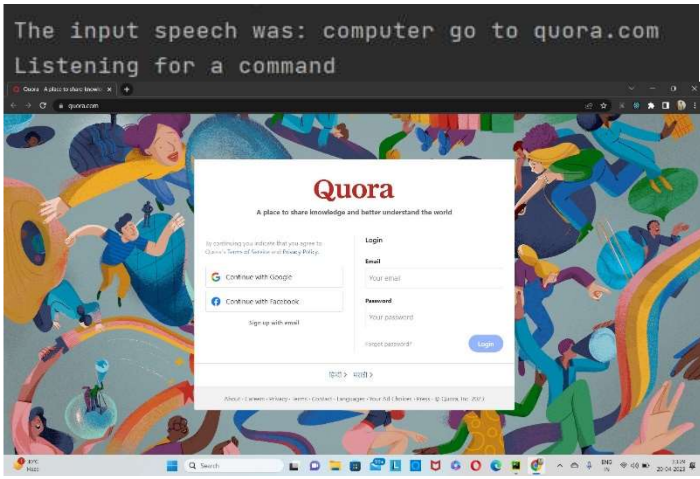
The project can be extended to integrate with IoT (Internet of Things) devices such as smart home appliances and devices, such as lights, thermostats, and security systems. This integration would allow users to control their home devices using voice commands.

VII. RESULTS AND OBSERVATIONS



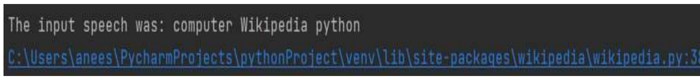
```
C:\Users\anees\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\anees\PycharmProjects\pythonProject\a
Start your command with:
 ACTIVATION WORD : COMPUTER IS A MUST TO INCLUDE IN A COMMAND.
 1. Say - Jarvis repeats whatever you said.
 2. Say Hello - Jarvis wishes you greetings.
 3. Go to - Jarvis redirects you to whatever website you say.
 4. Wikipedia - Jarvis searches for whatever you say in wikipedia and reads out the summary.
 5. Compute - Jarvis will perform computations such as calculating the distance between 2 cities, countries, etc
 6. Log - Jarvis takes note of what you say, makes a new file and writes the recorded part there.
 8. Exit - Jarvis says good bye, and the code ends.
 7. GPT - Uses OpenAI to query any question.
Listening for a command
```

Result 1: Initialisation of text to speech.



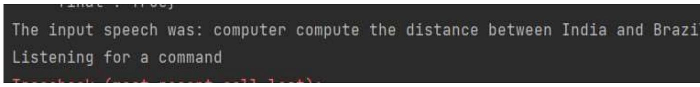
```
The input speech was: computer go to quora.com
Listening for a command
```

Result 2: Navigation to any website using assistant.



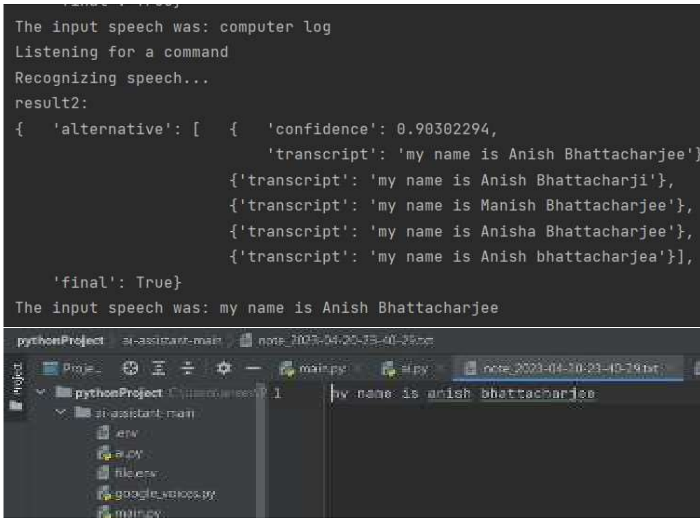
```
The input speech was: computer Wikipedia python
C:\Users\anees\PycharmProjects\pythonProject\venv\lib\site-packages\wikipedia\wikipedia.py:3
```

Result 3: Querying Wikipedia.



```
The input speech was: computer compute the distance between India and Brazi
Listening for a command
```

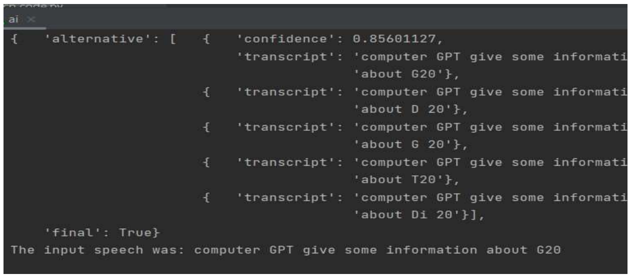
Result 4: Querying WolframAlpha.



```
The input speech was: computer log
Listening for a command
Recognizing speech...
result2:
{   'alternative': [   {   'confidence': 0.90302294,
                           'transcript': 'my name is Anish Bhattacharjee'}
                       {'transcript': 'my name is Anish Bhattacharji'},
                       {'transcript': 'my name is Manish Bhattacharjee'},
                       {'transcript': 'my name is Anisha Bhattacharjee'},
                       {'transcript': 'my name is Anish bhattacharjea'}],
    'final': True}
The input speech was: my name is Anish Bhattacharjee
```

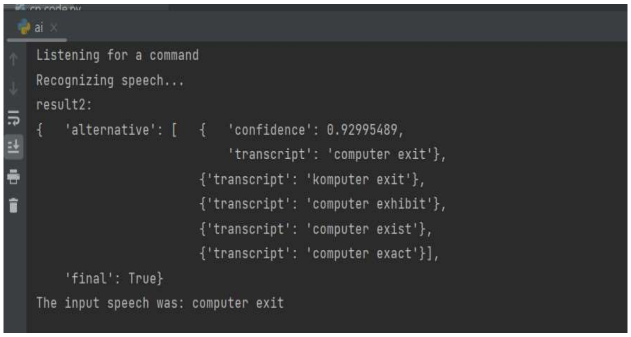
```
1    my name is anish bhattacharjee
```

Result 5: Logging command using the assistant.



```
{   'alternative': [   {   'confidence': 0.85601127,
                           'transcript': 'computer GPT give some informati
                                         'about G20'},
                       {   'transcript': 'computer GPT give some informati
                                         'about D 20'},
                       {   'transcript': 'computer GPT give some informati
                                         'about G 20'},
                       {   'transcript': 'computer GPT give some informati
                                         'about T20'},
                       {   'transcript': 'computer GPT give some informati
                                         'about Di 20'}],
    'final': True}
The input speech was: computer GPT give some information about G20
```

Result 6: Querying GPT



```
Listening for a command
Recognizing speech...
result2:
{   'alternative': [   {   'confidence': 0.92995489,
                           'transcript': 'computer exit'},
                       {'transcript': 'komputer exit'},
                       {'transcript': 'computer exhibit'},
                       {'transcript': 'computer exist'},
                       {'transcript': 'computer exact'}],
    'final': True}
The input speech was: computer exit
```

Result 7: Exiting the voice assistant.

VIII. CONCLUSION

In this paper we have discussed the uses, methodology as well as implementation details of an Artificial Intelligence Voice assistant using Python, text-to-speech, and GPT. The project involves integrating several libraries and APIs, including the webbrowser module, Wolfram Alpha client, SpeechRecognition, Pygame, and OpenAI's GPT. The voice assistant can recognize speech input and perform various tasks, such as searching Wikipedia and Wolfram Alpha, answering questions, and generating responses using OpenAI's GPT. The project also includes a text-to-speech feature that allows the voice assistant to respond verbally to user queries. Overall, the project demonstrates a strong understanding of Python programming and various AI-related APIs and technologies. The development of this AI voice assistant could have numerous practical applications, from personal use to commercial products.

ACKNOWLEDGMENTS

As the co-authors of this research paper titled "Artificial Intelligence Voice Assistant Using Python, Text to Speech and GPT," we would like to express my heartfelt gratitude towards everyone who has supported us throughout this journey.

We would like to thank our guide, Professor Swati Shamkuwar, for her unwavering support and guidance throughout this project. Her expertise in this field has been invaluable, and her feedback and suggestions have helped us improve the quality of our research.

Furthermore, we would like to thank my family and friends for their constant encouragement and support. Their unwavering belief in us has been a driving force throughout this journey.

Once again, we would like to extend my heartfelt gratitude to everyone who has played a part in making this research a success.

REFERENCES

[1] Mohasi, Lehlohonolo & Mashao, Daniel. (2006). Text-to-Speech Technology in Human-Computer Interaction. p. 3

[2] P. Nguyen, G. Heigold and G. Zweig, "Speech Recognition With Flat Direct Models," in IEEE Journal of Selected Topics in Signal Processing, vol. 4, no. 6, pp. 994-1006, Dec. 2010, doi: 10.1109/JSTSP.2010.2080812. p. 2

[3] Harshit Agrawal, Nivedita Singh, Gaurav Kumar, Dr. Diwakar Yagyasen, Mr. Surya Vikram Singh. "Voice Assistant Using Python" An International Open Access-revied, Refereed Journal.Unique Paper ID: 152099, Publication Volume & Issue: Volume 8, Issue 2, Page(s): 419-423. p. 2.

[4] Singh, Vivek. (2022). VIRTUAL ASSISTANT USING PYTHON. DOI: 10.13140/RG.2.2.28018.38088, p. 2-3.

[5] Kumar, D. R., Faisal, M., Faisal, M., & Ahmad, O. (2022). My voice assistant using python. *International Journal for Research in Applied Science and Engineering Technology*, *10*(4), 3004–3007. https://doi.org/10.22214/ijraset.2022.41958. p. 3

[6] G. Preethi , Abishek. K , Thiruppugal S , Vishwaa D A, 2022, Voice Assistant using Artificial Intelligence, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 05 (May 2022). p. 3

[7] Pooja C. Goutam,Monika S. Jalpure,Akshata S. Gavade,Pranjali Chaudhary,A. V. Gundavade, "VOICE ASSISTANT USING PYTHON", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 6, pp.c801-c806, June 2022, Available at: http://www.ijcrt.org/papers/IJCRT22A6338.pdf p. 3

[8] Huang, Y. (2023). Research on the Development of Voice Assistants in the Era of Artificial Intelligence. *SHS Web of Conferences*, *155*, 03019. https://doi.org/10.1051/shsconf/202315503019 p. 3

[9] B. S. Atal and L. R. Rabiner, "A pattern recognition approach to voiced unvoiced-silence classification with applications to speech recognition,"

[10] Hill, J., Ford, W.R., & Farreras, I.G. (2015). Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations. *Comput. Hum. Behav.*, *49*, 245-250.

[11] A. Trivedi, E. K. Kaur, C. Choudhary, Kunal and P. Barnwal, "Should AI Technologies Replace the Human Jobs?," 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/INOCON57975.2023.10101202

[12] P. W. Jaronde, A. Vyas and M. Gaikwad, "A Survey on Energy Aware Cognitive Radio Network," 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2022, pp. 1-6, doi: 10.1109/PARC52418.2022.9726654.

[13] D. Manikkule and P. Jaronde, "Encapsulation of Full Adder Using 180nm CNTFET," 2019 9th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-19), Nagpur, India, 2019, pp. 1-6, doi: 10.1109/ICETET-SIP-1946815.2019.9092017.

[14] D. Manikkule and P. Jaronde, "Design of Decoder Using Ternary Inverter," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-3, doi: 10.1109/I2CT45611.2019.9033760.

[15] Gaikwad, M.J., Asole, P.S. and Bitla, L.S., 2022, January. Effective Study of Machine Learning Algorithms for Heart Disease Prediction. In 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC) (pp. 1-6). IEEE.

[16] Sadani, U., Bitla, L., Vairagade, R. and Ghule, V., 2022, April. Power and Delay Efficient Three-Input XOR/XNOR With Systematic Cell Design Methodology. In 2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-22) (pp. 1-5). IEEE.

[17] Vairagade, R., Bitla, L., Judge, H.H., Dharpude, S.D. and Kekatpure, S.S., 2022, April. Proposal on NFT Minter for Blockchain-based Art-Work Trading System. In 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 571-576). IEEE.

[18] Bramhane, L., Salankar, S., Gaikwad, M., & Panchore, M. (2022). Impact of Work Function Engineering in Charge Plasma Based Bipolar Devices. Silicon, 14(8), 3993-3997.

[19] Bahel, Vedant, and Mahendra Gaikwad. "A Study of Light Intensity of Stars for Exoplanet Detection using Machine Learning." 2022 IEEE Region 10 Symposium (TENSYMP). IEEE, 2022.

[20] Gaikwad, Mahendra, Milind Khanapurkar, and Sachin Untawale. "Recent development of nano-satellite constellation as iot communication platform." AIP Conference Proceedings. Vol. 2424. No. 1. AIP Publishing LLC, 2022.

[21] Bahel, Vedant, et al. "Supervised Classification for Analysis and Detection of Potentially Hazardous Asteroid." 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA). IEEE, 2021.