# AI based Voice Assistant

**Bachelor of Technology**

**Computer Science Engineering Department**

**CSDA**

**Netaji Subhas University of Technology, East Campus**

**Geeta Colony, Delhi**


**8th Semester**

Tusshar kashyap
(2020UCB6014)

Akash Sharma
(2020UCB6027)

Ashish Raj
(2020UCB6032)

MAY 2024

# CERTIFICATE

This is to certify that the project entitled **Al Voice Assistant** is the bonafide work of **Tusshar Kashyap (2020UCB6014), Akash Sharma (2020UCB6027) and Ashish Raj (2020UCB6032)**, who carried out the project work under my supervision. It has been duly presented in the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science Engineering (CSDA Branch)** at Netaji Subhas University of Technology, East Campus, Geeta Colony, Delhi. To the best of my knowledge, the thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Dr. Shobha Bhatt**

**Assistant Professor**

Department of Computer Science and Engineering,

Netaji Subhas University of Technology East Campus

Geeta Colony, Delhi - 110078

Signature :_____

Date :_____

## ACKNOWLEDGEMENT

We express our sincere thanks and deep sense of gratitude to our thesis supervisor, **Dr. Shobha Bhatt, Department of Computer Science and Engineering (East Campus),** Netaji Subhas University of Technology, for her invaluable motivation and guidance, without which this report would not have been possible. We are truly fortunate to have had the opportunity to learn and work under her expert supervision and guidance throughout our association with her. We hold deep admiration for her inherent kindness and academic prowess. Our profound gratitude extends to **Professor Manoj Kumar**, Head of the Computer Science & Engineering Department (East Campus), and **Professor Nanhay Singh** for providing the necessary facilities and support to conduct our research. We also owe a deep debt of gratitude to all our faculty members and lab staff for their insightful suggestions and for sharing their extensive experience with us. Additionally, we would like to extend our special thanks to **Prof. Arvind Kumar and Prof. Prakash Rao** for their support and contributions to our project. Their assistance has been invaluable, and we are grateful for his involvement.

| | | |
|---|---|---|
| **Tusshar Kashyap** | **Akash Sharma** | **Ashish Raj** |
| **(2020UCB6014)** | **(2020UCB6027)** | **(2020UCB6032)** |

# Abstract

This project focuses on the creation and implementation of an AI-driven voice assistant capable of executing a range of tasks, including greetings, news aggregation, mathematical computations, provision of random facts, music and video recommendations, and information retrieval. The voice assistant harnesses cutting-edge technologies like Python, speech recognition libraries, natural language processing tools, and external APIs to deliver an interactive and versatile user experience.

The greeting feature initiates user interaction by offering a warm welcome upon activation. The news aggregation capability enables users to receive up-to-date news headlines from reputable sources, ensuring they stay informed effortlessly.

Mathematical computations are performed accurately based on user input, providing a convenient tool for solving arithmetic problems. The random facts feature enhances user engagement by delivering intriguing pieces of information on diverse topics.

Additionally, the music and video recommendation functionality utilizes advanced algorithms to suggest personalized content from popular streaming platforms, catering to individual preferences.

The voice assistant's information retrieval feature empowers users to access detailed information on specific subjects, leveraging a variety of online resources such as Wikipedia and Wolfram Alpha. The project also provides the option to integrate a user-friendly graphical interface for enhanced accessibility.

Thorough testing and evaluation ensure the voice assistant's reliability and functionality across its diverse range of features. In conclusion, this project contributes to the advancement of AI-driven voice assistants, offering users a comprehensive and intuitive tool for various tasks and input.

# TABLES OF CONTENT

# List of Figures

# CHAPTER 1

# INTRODUCTION

The way people interact with information and access services has changed significantly in the modern digital age as a result of artificial intelligence (AI) being incorporated into commonplace technologies[2][3]. This paper aims to explore the development and integration of an AI voice assistant tailored to meet diverse user needs, including information provision, weather updates via API integration, calculator functionalities, and personalized music or audio suggestions[1].

## 1.1 Background

As technology continues to progress, the demand for intelligent and personalized solutions that enhance user experience has become increasingly prevalent. An important step forward in this evolution is the usage of artificial intelligence (AI) voice assistants, which uses Natural Language Processing and Machine Learning to understand user questions and provide conversational responses. By harnessing these capabilities, our AI voice assistant endeavors to provide users with an efficient and enjoyable interaction experience across multiple domains[3].

The fundamental functionalities of the AI voice assistant encompass serving as an information provider, utilizing its ability to fetch and present relevant data from various sources in response to user inquiries. Furthermore, integration with weather APIs empowers the assistant to deliver real-time weather updates customized to the user's location, thereby enhancing convenience and utility[6]. Additionally, the inclusion of a calculator feature enables users to execute mathematical calculations swiftly and effortlessly using voice commands, catering to productivity needs. Moreover, acknowledging the significance of personalized recommendations in augmenting user engagement, our AI voice assistant incorporates sophisticated recommendation algorithms to suggest music or audio content based on individual preferences and contextual relevance. By analyzing user interactions and feedback, the assistant continuously enhances its recommendations, aiming to provide a tailored and enjoyable audio experience[8][9].

The given Figure1.1 shows a detailed working of speech synthesis process, showcasing the various steps involved from input voice signal to final text output/voice output.

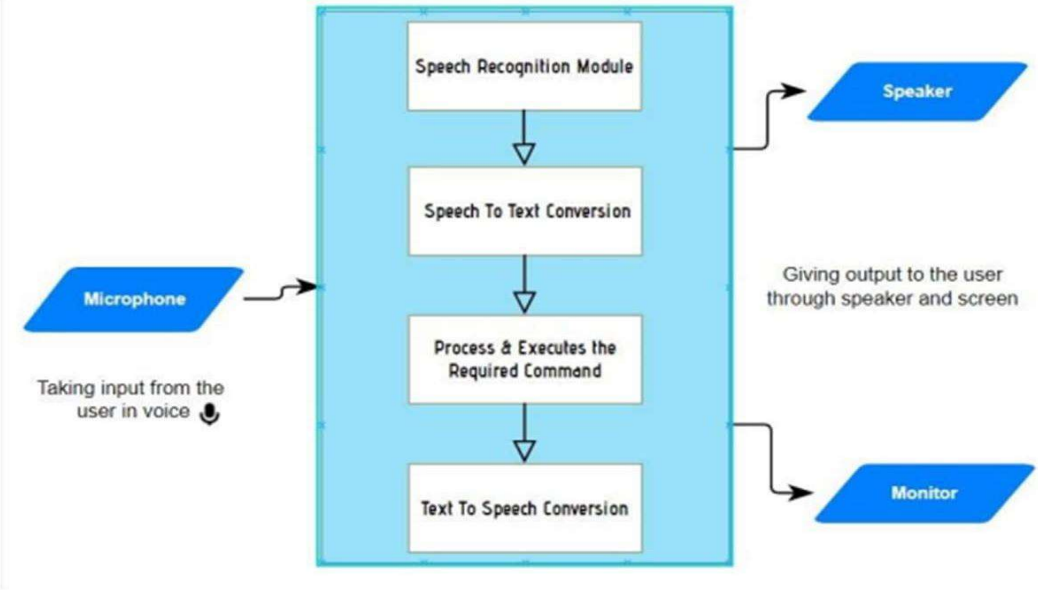


**Figure1.1: Working of Speech Synthesis**

## 1.2 Problem Statement

- Enhancing Weather API Integration: What methods can be employed to enhance the integration of weather APIs into AI voice assistants, ensuring real-time updates that are accurate, location-specific, and reliably delivered to users?

- Improving Calculator Functionality: How can the functionality and usability of the calculator feature within AI voice assistants be improved to support a broader range of mathematical operations and provide intuitive user interfaces?

- Personalizing Music or Audio Recommendations: What algorithms and approaches can be utilized to enhance the personalization of music or audio recommendations by AI voice assistants, considering user preferences, contextual relevance, and diversity of content?

### 1.2.1 Problem Identification

- Information Overload: With the abundance of information available online, users often struggle to sift through and find relevant information efficiently, leading to information overload and decreased productivity[4].

- Inaccurate Weather Forecasts: Existing weather forecast services may provide inaccurate or outdated information, leading to inconvenience and potential disruptions in users' plans and activities[7][8].

- Complex Calculations: Users may encounter difficulties in performing complex mathematical calculations manually, especially in time-sensitive situations where quick and accurate results are required[5][6].

- Generic Recommendations: Current recommendation systems employed by AI voice assistants may offer generic suggestions for music or audio content, failing to cater to individual preferences and resulting in user dissatisfaction[1][2].

## 1.3 Motivation

The motivation behind developing and integrating an AI voice assistant with capabilities such as information provision, weather updates via API integration, calculator functionalities, and personalized music or audio suggestions stems from the desire to enhance user experience and convenience in an increasingly digital world. First off, the growth of digital information has made efficient and user-friendly data access and processing techniques necessary. An artificial intelligence (AI) voice assistant can function as an information provider by utilizing machine learning and natural language processing (NLP) methods[5]. This allows users to conveniently and rapidly access pertinent data from several sources by using voice commands. Secondly, real-time weather updates are essential for users to plan their activities and make informed decisions. Integrating weather APIs into the AI voice assistant allows for the seamless delivery of weather information tailored to the user's location, enhancing convenience and utility. Thirdly, in today's fast-paced environment, users often require quick access to basic computational functionalities such as calculations[7]. By incorporating a calculator feature into the AI voice assistant, users can perform mathematical tasks efficiently using voice commands, eliminating the need for manual input and streamlining the workflow[9].

## 1.4 Research Gap

The subtle interactions between Natural Language Processing and Speech Recognition accuracy, especially in noisy real-world settings, are still poorly understood, despite the advances in AI voice assistants that use libraries like pyttsx3 and Speech Recognition. Existing studies often focus on the technical implementation and performance metrics of these libraries but overlook the complex linguistic and environmental factors that influence the user experience[8].

- **Some potential research directions to bridge this gap include:**

  Strength in Noisy Environments: Researching methods to make AI voice assistants more resilient to outside noise, such as background conversations, music, or other acoustic disruptions. This could involve exploring advanced signal processing algorithms or integrating contextual awareness to adaptively filter out noise[4].

- **Accent and Dialect Adaptation**:

  Understanding the challenges associated with diverse accents and dialects in speech recognition, especially for users from non-standard linguistic backgrounds. Research could focus on developing accent-adaptive models or dialect-specific training data to improve recognition accuracy and user satisfaction[7].

- **Comprehending Semantics and Contextual Knowledge:**

  Examining how AI voice assistants incorporate contextual awareness and semantic comprehension to enable more intuitive and natural interactions. This procedure entails investigating methods from Dialogue Management and Natural Language Understanding in order to deduce user intent and sustain cohesive talks over several turns[6].

- **User-Centric Design and Evaluation**:

  Shifting the focus from purely technical benchmarks to user-centric design and evaluation methodologies. This could include conducting user studies to assess usability, satisfaction, and acceptance of AI voice assistants in real-world scenarios, considering factors such as user demographics, task complexity, and usability barriers[9].

- **Ethical and Societal Implications**:

  Analyzing the social and ethical ramifications of AI voice assistants being widely used, taking into account concerns about algorithmic unfairness, data security, and privacy. Research in this area could involve developing frameworks for responsible AI design and governance to mitigate potential risks and ensure equitable access to voice-based technologies[2].

### 1.4.1 Research Objectives

1. **To perform an overload of data**: With the abundance of information available online, users often struggle to sift through and find relevant information efficiently, leading to information overload and decreased productivity.

2. **To Implement General Advice**: Current recommendation systems employed by AI voice assistants may offer generic suggestions for music or audio content, failing to cater to individual preferences and resulting in user dissatisfaction.

3. **To take action on security and privacy concerns**: Implement stringent privacy and security measures within the AI voice assistant to safeguard user data and ensure confidentiality, building trust and confidence among users.

4. **In order to improve accessibility**: Develop features and functionalities within the AI voice assistant to improve accessibility for users with disabilities or language barriers, ensuring inclusivity and equal access to the technology.

5. **To Completely Enhance Offline Features**: Explore strategies to enhance offline functionality within the AI voice assistant, minimizing dependency on internet connectivity and improving usability in areas with limited or unreliable internet access

## 1.6 Summary

The above Introduction cover a range of topics and developments related to AI voice assistants, with an emphasis on improving the features and usability of these assistants for a variety of purposes, including informational purposes, weather reports, calculator functions, and customized audio recommendations[7]. This is a comprehensive overview:

1. **Functions and Improvements of the AI Voice Assistant**: The AI voice assistant is a versatile tool that can be used to retrieve data, carry out computations, get the weather, and get recommendations for songs based on your tastes. The goal is to incorporate sophisticated features including improved algorithms for personalized audio suggestions, a powerful calculator for a wide range of mathematical operations, and a weather API for real-time, location-specific updates[5].

2. **Challenges and Problem Identification**: Managing information overload, providing precise weather forecasts, enabling intricate computations, and avoiding general advice are the main problems that have been found. Improving the functionality and user experience of AI voice assistants requires addressing these issues

3. **Research Directions**: Further studies could focus on user-centric design and evaluation, adapting to different accents and dialects, improving semantic and contextual understanding, making AI voice assistants more effective in noisy environments, and investigating the social and ethical ramifications of widespread AI assistant use.

4. **Goals and Motivations**: The goals center on improving the AI voice assistant's capacity to effectively handle large amounts of data, deliver precise weather predictions and computations, and make tailored music suggestions. To maintain inclusivity and user confidence, a strong emphasis is also placed on privacy, security, accessibility, and dependable offline functionality.

To put it simply, the main objective is to create AI voice assistants that improve engagement and utility in a digitally linked world by addressing practical usability, ethical standards, and user-centered design in addition to meeting technical requirements[6][7][9].

# Chapter 2
# Related Literature Study

Artificial intelligence voice assistants have become a key breakthrough, improving daily interactions and information accessibility as AI continues to redefine technological boundaries. This chapter explores the corpus of research on AI voice assistants, emphasizing their creation, features, dynamics of user engagement, and the technology underpinnings that enable them to function. This study attempts to contextualize the development of voice assistant technologies, identify current gaps, and provide guidance for the creation of more sophisticated and user-centric AI voice assistants by looking at related literature.

### 2.1 Technical Review of Speech Recognition and Text-to-Speech (TTS) Libraries

Explore existing literature that evaluates the performance, accuracy, and computational efficiency of speech recognition and TTS libraries like Speech Recognition and pyttsx3. Look for comparative studies that assess the strengths and limitations of different libraries in various contexts and applications[1][5][9].

## Speech Recognition Libraries

- **Google Text-to-Speech**

  Description: This robust API turns spoken utterances into text by leveraging Google's cutting-edge machine learning methods. It provides real-time streaming transcription and covers more than 120 languages and dialects[2][6].

  Advantages: Excellent precision, compatibility with multiple languages, resilience in cacophonous settings.

  Cons: You need to have internet access and pay for this service. The processing of data on Google's servers may give rise to privacy issues.

- **Microsoft Azure Speech to Text**

  Description: This tool, which is a component of Microsoft's Azure Cognitive Services, recognizes speech in real time and may be tailored to comprehend particular vocabularies or speaking styles[8][9].

  Advantages: provides wide language support, customized speech models, and real-time transcription.

  Limitations: Requires an Azure membership; due to cloud operation, latency and privacy issues may arise.

## Text to Speech Libraries

- **Polly from Amazon**

  Description: Polly, a feature of AWS, uses deep learning to transform text into realistic speech. It offers many different languages and voices[4][7].

  Advantages: Supports pitch, speed, and loudness adjustments using Speech Synthesis Markup Language (SSML). streaming in real time.

  Limitations: cloud-based, not free, requires AWS account (privacy problems).

- **Text-to-Speech on Google**

  Description: Uses deep learning models to translate text into speech that sounds natural. provides a variety of voices in many languages[2][6].

  Strengths: Excellent voices that accommodate a variety of dialects and languages.

  Limitations: Compared to other TTS systems, there are less customizing choices, and an internet connection is necessary for optimal results.

## 2.2 Review on Accent and Dialect Recognition

In the field of speech recognition technology, accent and dialect recognition in AI voice assistants is a crucial topic of research. The difficulties of processing and comprehending spoken language from people with different linguistic origins are addressed by this area of AI development. To make voice assistants more inclusive and useful for a wider audience, it is essential to recognize accents and dialects accurately[3][7].

**Challenges:**

1. Variability in Speech Patterns: Even within the same language, accents and dialects can differ greatly, impacting word usage, pronunciation, and intonation[2][4].

2. Restricted Training Data: The majority of speech recognition algorithms perform worse on minority accents and dialects since they were largely trained on data from dominant language groups[5][9].

3. Contextual Understanding: Natural language understanding can be hampered by accents and dialects because they may contain peculiar colloquialisms or idiomatic terms that are foreign to AI systems[3][7].

**Approaches:**

1. Deep Learning Models: The resilience of voice recognition systems against accent fluctuations has been greatly enhanced by advanced neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs)[5].

2. Transfer Learning: To improve understanding of other accents, a model trained on data from one is modified. Transfer learning aids in optimizing the model for particular, underrepresented accents while utilizing the vast datasets that are already available[8].

3. End-to-End Learning: In order to minimize the potential accent bias found in older phoneme-based systems, contemporary methods employ end-to-end models that translate voice to text directly without the need for intermediary phonetic representations[4][9].

4. Dialect-Aware Training: To improve recognition accuracy, include dialect-specific data in the training set or even build distinct models for each dialect[1][7].

## 2.3 Summary

In conclusion, these subjects taken as a whole shed light on the many facets of AI voice assistant development, including technological applications, sociocultural factors, and theoretical underpinnings. Resolving issues with accent and dialect detection, comprehending speech recognition and TTS technologies, and incorporating information from related literature are essential for improving the functionality and user experience of AI voice assistants in a variety of settings and areas[2,3,5].

# Chapter 3
# Methods and Methodology

To efficiently implement a personal voice assistant, we leverage the capabilities of speech recognition libraries coupled with various APIs for extended functionalities. Our approach integrates speech recognition functions to interpret user commands and deliver responses through text-to-speech functionality[6].figure 3.1 present a flow chart detailing the process of how our voice assistant takes user input and delivers output[7].

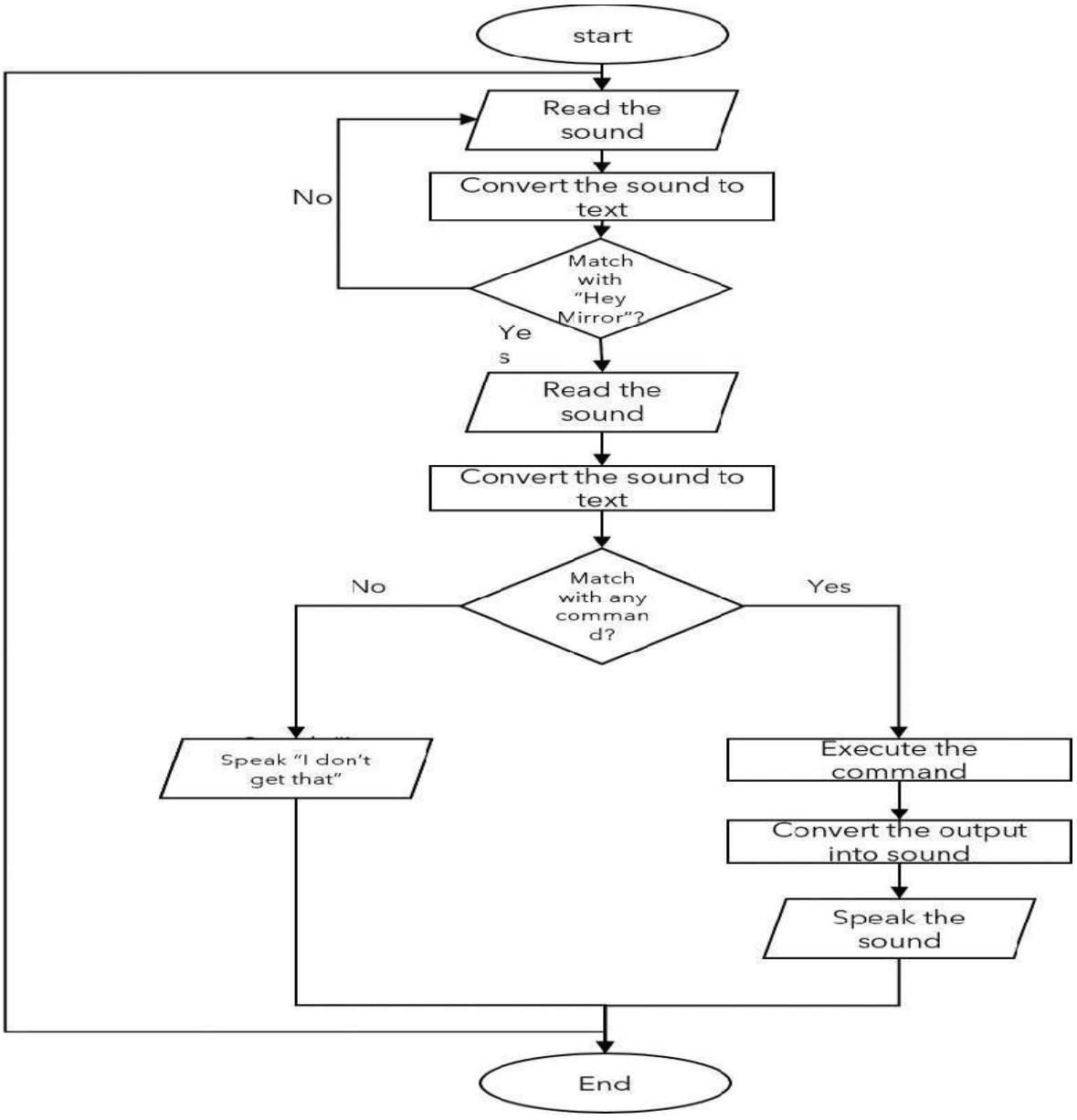


**Figure 3.1 Interaction Sequence Diagram / Framework**

We utilize existing libraries such as SAPI5 and pyttsx3 to harness system voices and enable offline text-to-speech conversion in Python. SAPI5, developed by Microsoft, facilitates speech recognition and synthesis within Windows applications, while pyttsx3 offers offline text-to-speech conversion compatible with both Python 2 and 3[7][8].

In order to vocalize program outputs, we define a talk function. We also construct a function that uses the system microphone to record voice commands. All program functionality is contained in the main function, which coordinates the smooth relationship between user commands and the intended activities made possible by APIs[9].

Apart from the basic features, we also include APIs for a variety of tasks including making calculations, getting news from the internet, and getting the weather. By sending requests and processing API responses, our assistant delivers relevant outputs tailored to user commands. This comprehensive approach ensures the personal voice assistant effectively interprets user inputs, executes desired actions, and communicates responses seamlessly, enhancing user experience and utility[3].

Figure 3.2 shows how our ai voice assistant takes input and with the help of different python libraries and Api gives us relevant output .

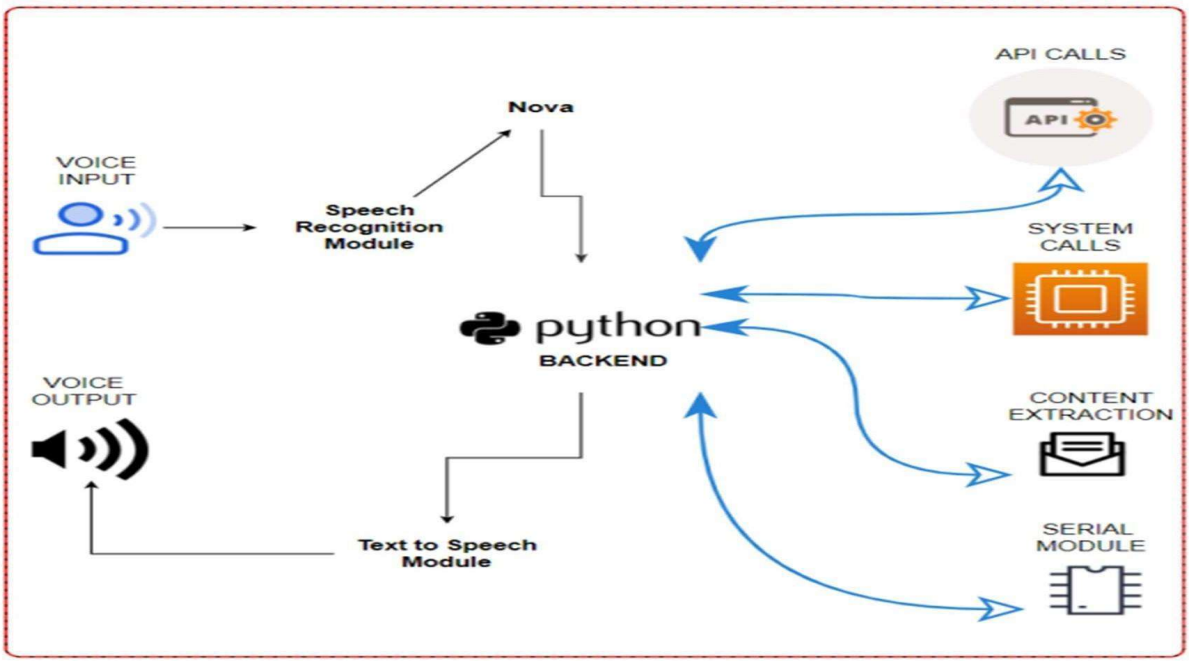


**Figure 3.2 System Architecture**

## 3.1 Algorithm Used

- **Speech Recognition Module**

- The class that we use is named Recognizer; it uses a module to provide the output in speech and turns the audio files into text[5].
- The energy level threshold for sounds is represented by the energy threshold function. those over this level are seen as speech, and those below it are regarded as quiet[7].
- The Recognizer instance uses audio from source (an Audio Source instant) to dynamically modify the energy threshold in order to accommodate for ambient noise (source, duration = 1)[5].

- **Speech To Text & Text to Speech Conversion**

- The Python text-to-speech conversion library is called Pyttsx3. and can use particular commands to adjust the Voice, Rate, and Volume. We have included sapi5 and espeak TTS Engines that can handle the same[3].
- Python offers an API called Speech Recognition that allows us to convert audio into text for further processing. Converting large or long audio files into text using the Speech Recognition API in Python[4].

- **Process & Execute Required Command**

- The command is transformed into text using a speech recognition module and thereafter saved in a temporary file. Next, use temp to analyze the user's words, determine the user's wants based on the input they have given, and execute the while loop. Next, commands are carried out[8].

## 3.2 Types of Operation

- **Information:** When we ask for information, Wikipedia opens and asks us what topic we want to learn more about. Then, it uses its xpath to click on the Wikipedia search box, looks up the topic there, clicks the button to activate the search, and reads a paragraph about it.

- **Plays the video which we ask**: When we ask it to play a video, it launches YouTube and prompts us to enter the name of the file. Subsequently, it utilizes its xpath to click on the YouTube search box, click the search button, and click the first search result using the xpath of the first video.

- **Fact**: It provides the user with a fact if they ask for a logical fact.

- **Temperature and Weather:** The current temperature is provided to the user upon request.

- **Mathematics Calculations:** If the user provides some integers for division, addition, and subtraction operations.

- **Date and Time**: Calculating time differences, Adding or subtracting time from dates Converting between different date and time formats

- **Self Introduction**:It will give proper information about itself like who made it ,its name and the purpose its making

## 3.3 Libraries Required

1. **Pyttsx3**: Pyttsx3 is a Python text-to-speech module that works with Python 2 and Python 3. It makes text to speech accessible for a range of applications by offering a straightforward user interface. A reference to a pyttsx3 engine can be obtained by applications by calling the pyttsx3.init() factory function[2].

2. **Speech Recognition**: Speech recognition is a crucial technology that enables computers to understand human language by identifying spoken words. In Python, the Speech Recognition library facilitates the conversion of spoken words into text, allowing for queries, responses, and various interactions. The Google voice Engine and Google Cloud Speech API are examples of supported voice recognition engines and APIs. Installing Speech Recognition is as simple as running pip install Speech Recognition[4][5].

3. **Rand-Facts**: Use the Python package Rand-facts to generate random facts, which will bring a playful and intriguing touch to programs. Using randfacts.get_fact(), developers can retrieve a random fun fact programmatically. This library is installed effortlessly via pip install rand-facts[1][8].

4. **Wikipedia**: The Wikipedia library for Python simplifies access to and parsing of data from Wikipedia, the multilingual online encyclopedia. With this library, users can search Wikipedia, retrieve article summaries, extract data such as links and images from pages, and more. Installation is done with pip install Wikipedia[7][8].

5. **Selenium WebDriver:** Selenium WebDriver is instrumental for automating web browser interactions in Python applications. It supports various browsers/drivers such as Firefox, Chrome, and Internet Explorer, as well as the Remote protocol. This versatility makes Selenium WebDriver a valuable tool for tasks requiring web automation. It's compatible with Python versions 3.5 and above, and installation is accomplished using pip install selenium[3][6].

6. **Date Time**: It's a standard python library which is essential for programming tasks that involves scheduling, timestamps, durations or period of time[9].

7. **WolfRamAlpha**: Wolfram Research has built a computational knowledge engine called Wolfram Alpha. Stephen Wolfram introduced it in 2009, and since then, it has grown in popularity as a resource for people looking for answers in a variety of fields, such as science, math, engineering, technology, business, art, and geography[4][6].

## 3.4 Summary

Our AI voice assistant's system architecture makes use of pre-existing libraries for text-to-speech (TTS) conversion and speech recognition, such as SAPI5 and pyttsx3, which makes offline Python functionality possible. Microsoft's SAPI5 makes speech synthesis and recognition easier for Windows apps, and PyTTSX3 provides cross-version compatibility for TTS conversion. We define two functions in our implementation: one for recording voice instructions via the system microphone, and the other for vocalizing program results. The primary function synchronizes API interactions with user commands to guarantee that intended activities are carried out without a hitch. Numerous functions are available from our assistance, such as getting news, performing arithmetic, and updating the weather. It improves user experience and utility by making requests, processing API replies, and delivering customized outputs in line with user directions[2][4][7]

# Chapter 4
# Results and Discussions

Our team set out to build a flexible and intelligent system that could improve user experience in a variety of tasks and areas in order to construct an AI voice assistant. This introduction provides a summary of our work to date, highlighting the major outcomes and accomplishments during the project's duration.

## 4.1 Information Result

When we ask our AI voice Assistant to give us information about Black hole, it take this as a command and redirected us to the Wikipedia site to give us relevant information about the topic we had asked. In Figure 4.1 shows the output of the information we asked to our AI voice assistant.

```
150
listening
hello how r u
listening
listening
```

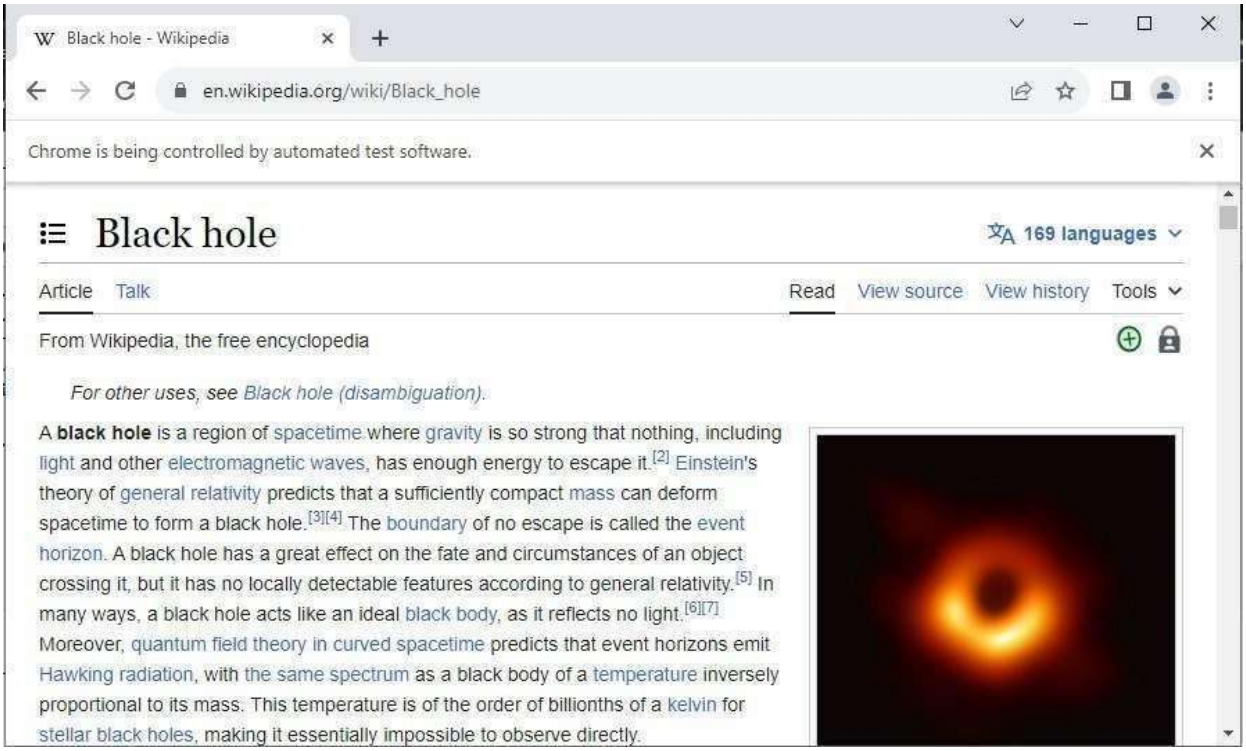


**Figure 4.1 Information Result about Black hole**

## 4.2 Mathematics Calculations

When we ask our AI voice Assistant to solve the simple calculation like we said to multiply two numbers then our voice assistant provide us the correct solution about the numbers we had asked. Figure 4.2 shows the calculation result of the question 930 x 262.

```
150
listening
hello
listening
calculate 930 X 262
243660
```

**Figure 4.2 Calculation Result**

## 4.3 Video Result

When we ask our AI voice Assistant to play a song or a video, it take this as a command and redirected us to YouTube to play the most related video about the topic we had asked. Figure 4.3 display the output of the video result as we pass the song name as a input query.

```
150
listening
hello how r u
listening
listening
```

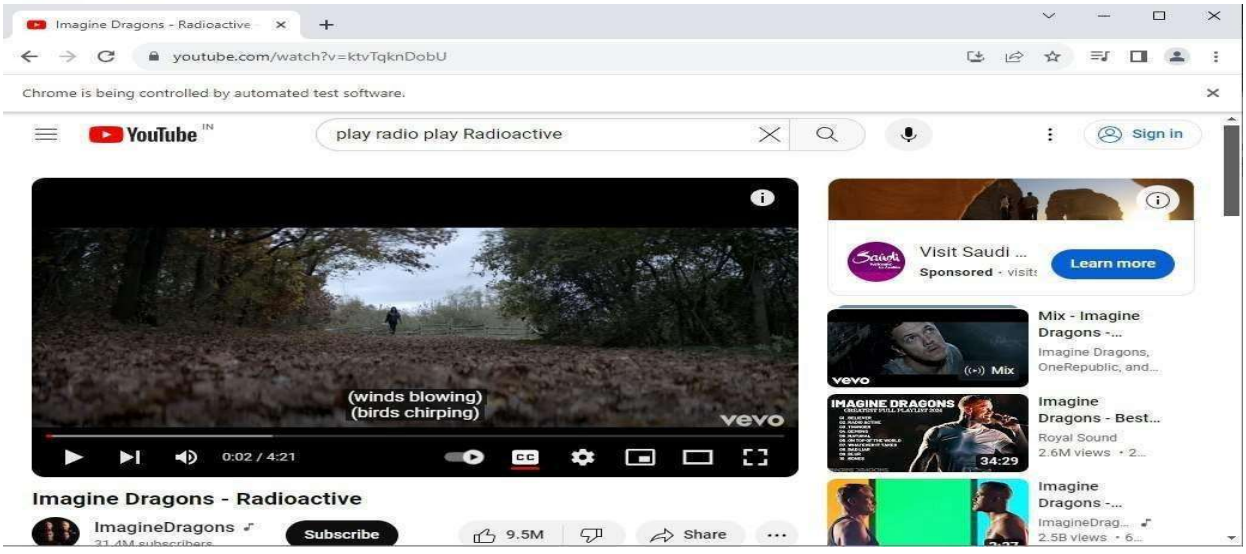


**Figure 4.3 Relevant video Result**

## 4.4 Random Facts

With the help of python library randfacts our ai Voice Assistant give us random facts whenever we asked it. Figure 4.4 display the random facts as we gave the query to our ai voice assistant.

```
150
listening
I am fine how are you
listening
C:\Users\Vikas\AppData\Local\Temp\ipykernel_10104\2448971805.py:104: DeprecationWarning: getFact is deprecated. Please use get_fact
  x = randfacts.getFact()
One billion seconds is about 32 years
```

**Figure 4.4 Relevant video Result**

## 4.5 NEWS Headlines

When we ask our ai voice assistant to gives us trendy NEWS or Headlines then it provides us the relevant information. Figure 4.5 shows the trending NEWS headlines as a text output when we gave our voice command as query.

```
150
listening
hello
listening
give me today's new give me today's news
1. Coronavirus in India live updates: Zydus' Virafin gets DCGI nod for Covid treatment
In yet another grim milestone, India recorded 3.3 lakh new Covid-19 cases, and 2,263 deaths in a day. Meanwhile, active cases crossed the 24-lakh mar

2. Government to provide 5 kg free food grains to poor for May & June
India News: The government on Friday announced to provide 5 kg free food grains to the poor for May and June 2021. This will cover nearly 80 crore benefi
ciaries u

3. PBKS vs MI Live Score, IPL 2021: Mumbai Indians seek consistency; Punjab Kings eye return to winning ways
IPL Live Score: Mumbai Indians seek consistency; Punjab Kings eye return to winning ways. Stay with TOI to get IPL live score, playing 11, scorecard, hig
hlights and ball by ball score updates of the 17th IPL match between Punjab Kings and Mumbai Indians.

4. Army explores procurement of 350 light tanks for mountainous terrain after border standoff with China
India News:  The Army is now exploring the possibility of procuring 350 light tanks, which can also be transported by air, to augment its firepower in hi
gh-altitu

5. 'Inappropriate': PM Modi objects to 'protocol break' during meeting; Delhi CM expresses regret
India News: Delhi chief minister Arvind Kejriwal on Friday faced flak for sharing a live telecast of an "in-house" meeting with Prime Minister Narendra M
odi where

6. Even record death toll may hide extent of India’s Covid crisis
India News: Bodies piling up at crematoriums and burial grounds across India are sparking concerns that the death toll from a ferocious new Covid-19 wave
may be m.
```

**Figure 4.5 NEWS Headlines Result**

## 4.6 Discussion

There is a signal word to begin it. For the same reason, users say the names of their voice assistants. They may shout "Alexa!" or maybe "Hey Siri!" The device is awakened by the signal word, whatever it may be. The voice assistant receives this as a cue to start listening. The voice assistant begins listening as soon as it hears its signal word. In order to determine whether your request is complete, the device waits for a pause. After that, the voice assistant routes our request to its source code. Our request is compared to other requests once it is in the source code. It's broken up into discrete orders that our voice assistant can comprehend.

The voice assistant receives these commands back from the source code. The voice assistant knows what to perform after it gets the commands. The voice assistant will perform the task we requested if it comprehends. As an example, "Hey, ROBO! DO YOU ACT IN A HEALTHY WAY? ROBO returns us report in a matter of seconds. The devices become more adept and responsive to human orders the more instructions they receive. The user uses a microphone to provide voice input, and the assistant responds by using the wake-up word to initiate Speech to Text (STT) and translating the voice input into text. It also comprehends the voice input and continues to perform the task that the user has repeatedly stated, delivering it via the AI-powered Text to Speech (TTS) module.

This project aims to develop and integrate an AI voice assistant with advanced capabilities including information provision, weather updates via API integration, calculator functionalities, and music or audio suggestions. The scope encompasses technological development, API integration, user interface design, privacy and security measures, accessibility features, and user testing.

The impact of this project lies in enhancing user experience and convenience across various domains, increasing efficiency and productivity, improving user satisfaction, ensuring inclusivity, and building trust and confidence among users regarding data privacy and security when interacting with the AI voice assistant. Ultimately, the project seeks to deliver a seamless and intuitive user experience, catering to diverse user needs and enhancing the overall usability and effectiveness of AI voice assistant technology.

# CHAPTER 5
# Conclusion and Future Study

An important turning point in human-computer interaction has been reached with the creation of AI voice assistants, which provide previously unheard-of levels of accessibility, efficiency, and convenience. This chapter outlines possible directions for further study and innovation in this rapidly evolving field while also reflecting on the outcomes and consequences of our AI voice assistant project.

## 5.1 Conclusion

As part of our research, we've created a sophisticated AI-powered voice assistant that can do a variety of functions, such as information retrieval, computations, weather forecasts, and audio/music suggestions. Our voice assistant shows how artificial intelligence (AI) has the potential to change human-computer interaction and simplify daily tasks with hands-free operation through the use of state-of-the-art natural language processing (NLP), machine learning, and seamless integration with external APIs technologies.

With rapid access to a plethora of knowledge on a wide range of topics, our voice assistant's information search feature enables users to quickly get pertinent information and improves user experiences. Furthermore, real-time forecasts and updates are provided by interaction with a weather API, enabling users to plan their activities and make educated decisions depending on the current weather.

The calculator feature makes it easy to perform mathematical operations with voice instructions, which increases efficiency and practicality in a variety of settings. Furthermore, by providing audio recommendations and carefully chosen music based on user tastes and mood, our voice assistant goes above and beyond being a useful tool to become an entertainment asset.

We faced difficulties with data pretreatment, complicated API integration, and performance optimization along the way. We did, however, overcome these challenges by working together and using creative problem-solving to accomplish our project's goals.

In the future, there will be many of chances to improve and grow our voice assistant driven by AI. In the future, it might be possible to enhance natural language understanding (NLU) to better understand user intents and context, expand language support, integrate voice-controlled smart home automation for added convenience, and improve the user interface for a more engaging and intuitive experience. With further innovation and improvement, our voice assistant will become an invaluable tool for users in a variety of fields.

## 5.2 Future Study

As we transition into an era driven by voice-activated technologies, it becomes imperative for businesses to adopt such innovations to maintain competitiveness and relevance. Voice activation not only enhances customer engagement but also streamlines internal operations, facilitating automation and saving valuable time.

1) **Sending Emails with a voice assistant**
   One significant application of voice-activated technology is in the realm of email communication. Emails serve as a cornerstone of professional correspondence, and leveraging a voice assistant to interact with email platforms like Gmail can expedite the process. Gmail, being a widely used and versatile email service offered by Google, offers seamless integration with voice-activated systems, enabling users to send and receive emails effortlessly.

2) **Scheduling appointments using a voice assistant:**
   Similarly, scheduling appointments can be significantly streamlined through voice-activated assistants. As businesses expand, the demands on time multiply, making efficient appointment management crucial. By utilizing voice commands, users can swiftly schedule appointments, check project updates, or arrange meetings with potential clients, thereby optimizing time management and productivity.

3) **Better Voice Assistant(VUI) Interface:** Enhancing the interface of voice assistants through Voice User Interfaces (VUIs) further augments their utility. VUIs enable users to interact with systems through spoken commands, freeing them from the constraints of manual input. Well-known virtual assistant such as Apple's Siri and Amazon's Alexa serve as prime examples of how VUIs may improve user experience enabling hands-free, eyes-free interaction, which in turn encourages multitasking and productivity.

# Refrences


1. Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends® in Signal Processing, 7(3–4), 197–387.

2. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82–97.

3. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681.

4. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2019). Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, 14(3), 55–75.

5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.

6. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

7. Rasa. (n.d.). Rasa: Open source conversational AI. Retrieved from https://rasa.com/

8. Mozilla. (n.d.). DeepSpeech: A TensorFlow implementation of Baidu's DeepSpeech architecture. Retrieved from https://github.com/mozilla/DeepSpeech

9. Python Software Foundation. (n.d.). Python programming language. Retrieved from https://www.python.org/

10. TensorFlow. (n.d.). TensorFlow: An open source machine learning framework for everyone. Retrieved from https://www.tensorflow.org/

# Appendix A
# Sample Code

```
import pyttsx3 as p
import speech_recognition as sr
from selenium import webdriver
from selenium.webdriver.common.by import By
import datetime
import wolframalpha
import json
import webbrowser
from urllib.request import urlopen
import requests
import os


class Email():
    def sendEmail(to, content):
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.ehlo()
        server.starttls()
        server.login('youremail@gmail.com', 'your-password')
        server.sendmail('youremail@gmail.com', to, content)
        server.close()

class Music():
    def __init__(self):
        self.driver = webdriver.Chrome()

    def play(self, query):
        self.query = query
        self.driver.get(url = "https://www.youtube.com/results?search_query=" + query)
        video = self.driver.find_element(By.XPATH, '//*[@id="video-title"]/yt-formatted-string')
        video.click()
```

```
class Infow():
    def __init__(self):
        self.driver = webdriver.Chrome()

    def get_info(self,query):
        self.query = query
        self.driver.get(url = "https://www.wikipedia.org")
        search = self.driver.find_element(By.XPATH,'//*[@id="searchInput"]')
        search.click()
        search.send_keys(query)
        enter = self.driver.find_element(By.XPATH,'//*[@id="search-form"]/fieldset/button')
        enter.click()

def WolfRamAlpha(query):
    apikey = "32X22R-JGKG6V62KY"
    requester = wolframalpha.Client(apikey)
    requested = requester.query(query)

    try:
        answer = next(requested.results).text
        return answer

    except:
        speak("The value is not answerable")
```

```
def Calc(text2):
    Term = str(text2)
    Term = Term.replace("robo","")
    Term = Term.replace("multiply","*")
    Term = Term.replace("plus","+")
    Term = Term.replace("minus","-")
    Term = Term.replace("divide","/")

    Final = str(Term)
    result = WolfRamAlpha(Final)
    try:
        result = WolfRamAlpha(Final)
        print(f"{result}")
        speak(result)

    except:
        speak("The value is not answerablr")


engine = p.init('sapi5')

voices = engine.getProperty('voices')

for voice in voices:
    if 'en' in voice.languages:
        engine.setProperty('voice', voice.id)
        break
def speak(text):
    engine.say(text)
    engine.runAndWait()
```

```
rate =engine.getProperty('rate')
print(rate)
engine.setProperty('rate', 150)

r = sr.Recognizer()

#speak("hello sir i am your voice assistant")
hour = int(datetime.datetime.now().hour)
if hour>=0 and hour<12:
    speak("Good Morning!")

elif hour>=12 and hour<18:
    speak("Good Afternoon!")

else:
    speak("Good Evening!")

speak("I am robo")
with sr.Microphone() as source:
    r.energy_threshold = 10000
    r.adjust_for_ambient_noise(source,1.2)
    print("listening")
    audio = r.listen(source)
    text = r.recognize_google(audio)
    print(text)
```

```
if "hello" and "are" and "you" in text:
    speak("i am fine")
speak("what can i do for you")
with sr.Microphone() as source:
    r.energy_threshold = 10000
    r.adjust_for_ambient_noise(source,1.2)
    print("listening")
    audio = r.listen(source)
    text2 = r.recognize_google(audio)

if "information" in text2:
    speak("you need information related to which topic?")


    with sr.Microphone() as source:
         r.energy_threshold = 10000
         r.adjust_for_ambient_noise(source,1.2)
         print("listening")
         audio = r.listen(source)
         infor = r.recognize_google(audio)

    speak("searching in wikipedia".format(infor))
    assist = Infow()
    assist.get_info(infor)

    with sr.Microphone() as source:
         r.energy_threshold = 10000
         r.adjust_for_ambient_noise(source,1.2)
         print("listening")
         audio = r.listen(source)
         text2 = r.recognize_google(audio)
         print(text2)
```

```
if "play" and "video" in text2:
    speak("you want me to play which video?")
    with sr.Microphone() as source:
         r.energy_threshold = 10000
         r.adjust_for_ambient_noise(source,1.2)
         print("listening")
         audio = r.listen(source)
         vid = r.recognize_google(audio)
    assist = Music()
    assist.play(vid)

if "time" and "date" in text2:
    print(text2)
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"sir, the time is {strTime}")
    with sr.Microphone() as source:
        r.energy_threshold = 10000
        r.adjust_for_ambient_noise(source,1.2)
        print("listening")
        audio = r.listen(source)
        text2 = r.recognize_google(audio)
        print(text2)

if "who made you" in text2 or "who created you" in text2:
    speak("My Creaters are Akash, Tusshar, Ashish.")


if 'exit' in text2:
    speak("Thanks for giving me your time")
    exit()
```

```
if 'reason for you' in text2:
    speak("I was created as a Minor project. ")

if "calculate" in text2:
    text2 = text2.replace("calulculate","")
    print(text2)
    Calc(text2)

if 'news' in text2:
    def get_latest_news():
        try:
            url = 'https://newsapi.org/v1/articles'
            params = {
                'source': 'the-times-of-india',
                'from': '2024-04-01',
                'sortBy': 'top',
                'apiKey': 'cb6f4ddba861453baa74efc603b3ab0d'
            }
            response = requests.get(url, params=params)
            response.raise_for_status()  # Raise an exception for HTTP errors
            data = response.json()

            if 'articles' in data:
                articles = data['articles']
                for i, article in enumerate(articles, 1):
                    print(f"{i}. {article['title']}")
                    print(article['description'] + '\n')
                    # You can add speak functionality here to read out the news
            else:
                print("No articles found in the response.")
        except requests.RequestException as e:
            print(f"Error fetching news: {e}")
```

```
    print(text2)
    speak("Todays news is")
    get_latest_news()
```

# wertt

ORIGINALITY REPORT

| 12% | 7% | 3% | 7% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to University of Bedfordshire<br>Student Paper | 4% |
| 2 | sist.sathyabama.ac.in<br>Internet Source | 3% |
| 3 | Submitted to Asia Pacific Instutute of Information Technology<br>Student Paper | 1% |
| 4 | Submitted to Buckinghamshire Chilterns University College<br>Student Paper | <1% |
| 5 | iajit.org<br>Internet Source | <1% |
| 6 | 5dok.org<br>Internet Source | <1% |
| 7 | Submitted to University of Aberdeen<br>Student Paper | <1% |
| 8 | lnu.diva-portal.org<br>Internet Source | <1% |
| 9 | Submitted to University of Central Lancashire | |

| | | |
|---|---|---|
| | Student Paper | <1% |
| 10 | R. Chaudhary, M. Kumar. "Hybrid Classifier for Crowd Anomaly Detection with Bernoulli Map Evaluation", International Journal on Artificial Intelligence Tools, 2024<br>Publication | <1% |
| 11 | biblio.ebookpoint.pl<br>Internet Source | <1% |
| 12 | www.lihpao.com<br>Internet Source | <1% |
| 13 | cdn.techscience.cn<br>Internet Source | <1% |
| 14 | Submitted to Netaji Subhas Institute of Technology<br>Student Paper | <1% |
| 15 | ethesis.nitrkl.ac.in<br>Internet Source | <1% |
| 16 | www.statista.com<br>Internet Source | <1% |
| 17 | theses.hal.science<br>Internet Source | <1% |
| 18 | www.grin.com<br>Internet Source | <1% |