

# Virtual Assistance using Python

**Pranav Ambhore, Pavan Ingole, Ritesh Shinde, Vedant Jadhav, Prof. S. L. Dawkhar**

Department of Information Technology  
Sinhgad College of Engineering, Pune, India

**Abstract:** *The paper investigates the potential of leveraging new technology to develop an intelligent Virtual Assistant that utilizes natural language processing and user-based data. It analyzes existing intelligent programs across various support categories and assesses the potential utility of a specific software as a Virtual Assistant. The envisioned Virtual Assistant is expected to possess social communication abilities through natural language processing, the capability to store and analyze user data, and the capacity to operate autonomously without human input or programming requirements. The paper suggests that advancements in technology may pave the way for the realization of virtual personal assistants. The authors conducted experiments on a particular software and conducted user testing, which demonstrated the feasibility of a basic program incorporating natural language processing algorithms. Overall, the paper presents the concept of an intelligent Virtual Assistant that has the potential to revolutionize our interaction with technology*

**Keywords:** The provided keywords relate to the development and technology associated with virtual assistants. They cover a wide range of topics, including the Python programming language, Natural Language Processing (NLP), the Wolfram Alpha API, desktop assistants, machine learning, text-to-speech, speech-to-text, language processing, voice recognition, artificial intelligence, Internet of Things (IoT), and virtual assistants. If you have a particular subject or query in mind, I'm here to assist you by leveraging these keywords.

## I. INTRODUCTION

The main goal of this project is to develop a user-friendly and self-contained application that assists individuals with limited technical skills or computer literacy in effectively using their computers. With the increasing ubiquity of computers in everyday life, this application aims to provide a solution for those who may find technology overwhelming.

The application operates similarly to popular virtual assistants like Siri and Google Assistant but focuses specifically on assisting users with personal computers. Users can interact with the virtual assistant using voice commands, keyboard inputs, or remote access via the internet.

The system includes a predefined set of commands and also allows users to create custom commands tailored to their specific needs. The virtual assistant can provide notifications for incoming emails, deliver weather information and location details, and even retrieve IP addresses, MAC addresses, and wireless network passwords. Furthermore, users can perform system tasks, such as shutting down or putting their computer to sleep, using voice commands.

Additionally, the application offers a note-taking feature. The system incorporates an interface that captures inputs and generates outputs. Different modules are available for various tasks like time management and searching, ensuring efficient execution of commands and tasks in a user-friendly manner. The ultimate objective is to simplify computer usage for individuals with limited technical skills, promoting ease of use and accessibility.

This project focuses on developing a Windows-based personal assistant that utilizes voice recognition technology. The application offers a wide range of services and functionalities, including mail sending, event handling, location services, music player support, weather checks, Google and Wikipedia search engines, camera integration, and compatibility with Bluetooth headsets. By incorporating these commonly used desktop services, this program has the potential to greatly enhance daily life and increase convenience. Additionally, it can provide valuable assistance to individuals with physical disabilities who may encounter difficulties with manual operations. The inspiration for this

program stems in part from Apple's popular virtual assistant, "Siri," which played a role in selecting it as the subject for a diploma project.

The launch of this application coincided with the debut of the iPhone4S, offering a highly engaging and user-friendly tool with extensive versatility and significant potential for future advancements. It caters to all generations and professional backgrounds, making it applicable across various industries. However, given its current exclusive availability on Apple's iOS, we have introduced a "Virtual Assistant" for personal computers to overcome this limitation. Voice assistance proves particularly valuable for personal assistants, navigation, driving, and aiding individuals with disabilities, among other scenarios.

According to Wikipedia, "Siri" is an intelligent personal assistant and knowledge navigator developed for Apple's iOS. It utilizes a natural language interface to respond to inquiries, provide recommendations, and execute actions by delegating requests to a collection of web services. Apple states that the software adapts to individual user preferences over time, delivering personalized results and accomplishing tasks such as identifying nearby restaurants and offering directions.

## II. LITERATURE SURVEY

Speech recognition technology has undergone significant advancements, making it a standard feature in smartphones and wearable devices. This technology enables users to dictate, search, and give voice commands. Research has been dedicated to designing a compact, efficient, and accurate large vocabulary speech recognition system for computers, particularly focusing on improving performance for Windows users accessing these services through portable devices. Techniques involving machine learning and natural language processing have been refined, leading to the emergence of applications that offer speech-based services and content. As computers become smaller and more pervasive, such as with wearables and Internet of Things (IoT) devices, speech recognition technology plays an increasingly vital role. The main purpose of a speech recognizer is to convert spoken words into text or other data formats.

One notable example of widely-used speech recognition application is "SIRI" on iPhones. SIRI acts as a voice-enabled personal assistant, allowing users to interact with their mobile devices through voice commands. SIRI processes user input and provides responses in various formats, such as executing actions or providing information. The proposed system aims to revolutionize the way Windows users engage with their devices, offering greater convenience and ease of use.

Cortana, a virtual assistant designed for desktop use, offers a wide range of functionalities such as message reading, location tracking, browsing history monitoring, contact list checking, and personalized suggestions based on the integrated data (with user permission). Alternatively, users have the option to type their queries or requests if they prefer not to use voice commands.

Siri, introduced with iOS 5 in 2011, started with basic capabilities like weather updates and messaging. However, it has since expanded significantly and now supports numerous third-party integrations on MacOS. While known for its humor, Siri has evolved to become more powerful over time. It can now make calls, send messages, schedule meetings, launch applications and games, play music, answer questions, set reminders, and provide weather forecasts.

Google Assistant stands apart from Cortana and Siri. It incorporates the functionalities of the earlier Google Now, which is currently being phased out.

### A. Background of the industry based

This project draws inspiration from Apple's widely-used application, "Siri," which was introduced alongside the iPhone4S. With its user-friendly interface and versatile functionalities, Siri has demonstrated its potential for various real-world applications and opportunities for further advancement, making it suitable for diverse industries. However, since Siri is exclusive to Apple devices, the need arose to develop a "Virtual Assistant" specifically for personal computers running on the Windows operating system. The integration of voice-assisted technology has proven beneficial in personal assistant roles, navigation aids, driving assistance, and supporting individuals with disabilities, among other applications. Siri, an intelligent personal assistant developed by Apple, operates as an iOS application that utilizes a natural language interface to provide answers, suggestions, and perform tasks by leveraging web services.

Apple asserts that the software continuously learns and adapts to user preferences, delivering personalized outcomes such as restaurant recommendations and directions.

### **B. Information Retrieval**

The program encompasses a diverse range of functions and services, catering to various needs and requirements. These include mail exchange, mathematical calculations, location services, music player functionality, weather checking, integration with Google and Wikipedia search engines, camera support, Bluetooth headset compatibility, IP address retrieval, finding saved Wi-Fi passwords, and a comprehensive help menu. Each function serves a specific purpose and operates based on specific instructions and criteria, as described below:

The mail exchange feature allows users to send emails to contacts in their address book. By issuing the appropriate command with the designated keywords and specifying the recipient, the program ensures that the intended person receives the email.

Location services provide users with the ability to check their current location or search for a desired destination. Depending on the request category, the program displays a user-friendly map with relevant locations, helping users navigate effectively.

The music player service empowers users to play specific songs or enjoy random selections from their stored song list on their desktop PCs. Users have the flexibility to pause or stop the music at any time, ensuring a personalized and immersive musical experience.

The weather checking function equips users with up-to-date weather information for any location. This includes essential details such as temperature and humidity. Additionally, users can retrieve weather forecasts for the current day, tomorrow, or the next four days, assisting them in planning activities accordingly.

The program integrates a Google search engine, enabling users to conduct searches and retrieve results directly within their web browser. By leveraging the power of Google's search capabilities, users can explore a wide range of topics and access relevant information conveniently.

Similarly, the program incorporates a Wikipedia search engine, granting users the ability to search for information within the vast Wikipedia encyclopedia. The searched content is presented directly in the browser, providing users with quick access to valuable knowledge.

Through its comprehensive range of features and services, the program aims to enhance the user experience by streamlining essential tasks and facilitating easy access to information.

The camera function within the program utilizes the mobile phone camera to capture an image of the current view. The captured image is then stored in the gallery, allowing users to access and manipulate it at a later time.

To address challenges with voice recognition, particularly when the music player is active or when the surroundings are noisy, the program offers support for Bluetooth headsets. This feature enables users to communicate with the headset instead of relying on the PC's microphone, enhancing the accuracy and convenience of voice commands.

Additionally, the program provides an application opening and closing service, granting users the ability to launch various installed applications such as MS Word, MS PowerPoint, Paint, Notepad, IDEs, Text Editors, File Explorer, and more. This functionality allows for seamless access to a wide range of software applications.

For programmers and IT personnel, the program incorporates an IP address service that facilitates the retrieval of the PC's hostname and IP address. Furthermore, it offers a MAC address finding service to assist users in obtaining their PC's MAC address.

A wireless network password finding service is also available, enabling users to retrieve saved WiFi passwords effortlessly. Additionally, the program includes a note writing service that utilizes a speech-to-text module, making it convenient to dictate and transcribe notes.

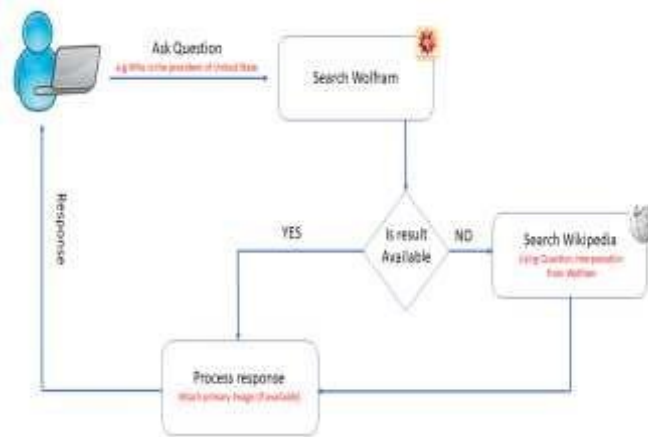
Entertainment is catered to with a dedicated joke service, providing users with humorous content. Additionally, the program offers a time and date service that can provide the current time, date, or day on demand. Moreover, it features a calculation service capable of performing various mathematical operations such as multiplication, division, and obtaining trigonometric values like sine and cosine.

These diverse services and features within the program aim to enhance functionality, convenience, and entertainment for users across different scenarios and requirements.

### C. Theory Model

This project encompasses a wide range of software engineering principles, software development models, Python programming skills, APIs, and network communication technologies. In particular, the focus is on the utilization of APIs and web services, with the Wolfram Alpha API being a key component. By leveraging this API, developers are able to streamline their workflow as the API handles the execution, eliminating the need for writing additional code. Consequently, the API serves as a crucial concept that guides the development theory of the project.

The Wolfram Alpha Web service API offers a web-based interface that enables seamless integration of Wolfram Alpha's powerful computational and presentation capabilities into various applications, including web, mobile, desktop, and enterprise solutions. By utilizing Wolfram's advanced algorithms, vast knowledgebase, and AI technology, the Wolfram Alpha API delivers expert-level answers to computational queries. The underlying Wolfram Language plays a pivotal role in enabling this functionality. In this article, step-by-step instructions are provided on how to develop a simple assistant application in Python that is capable of answering basic questions using the Wolfram Alpha API.



**Figure 1**

The input provided by the user will be forwarded to Wolfram Alpha for processing. Upon receiving a response from Wolfram Alpha, it will be returned to the user. However, if no response is received, the input will be treated as keyword(s) and utilized to perform a search query on Wikipedia.

## III. EQUIPMENT – CHOICE OF MATERIALS

### A. Tools and Environment Development:

**Python 3:** Python is a versatile, interactive, high-level, and object-oriented programming language. The release of Python 3.0 in 2008 introduced significant changes that rendered it incompatible with the earlier version. However, to ensure compatibility with Python 2.7, essential features were backported.

**Visual Studio Code:** Developed by Microsoft, Visual Studio Code is a source-code editor that offers multi-platform support for Windows, Linux, and macOS. It provides a range of features such as debugging support, integrated Git control and GitHub integration, syntax highlighting, intelligent code completion, code snippets, and refactoring capabilities.

**Sublime Text:** Sublime Text is a cross-platform source code editor available as shareware. It includes a Python API and supports various programming and markup languages. Users can enhance its functionality by adding community-built plugins, typically distributed under free-software licenses.

### B. API and Python Libraries:

**Google Speech API:** Speech recognition plays a crucial role in various applications, including home automation and artificial intelligence. This article introduces the SpeechRecognition library in Python, which offers convenient speech recognition capabilities. One notable advantage is its compatibility with microcontrollers like Raspberry Pi when used with an external microphone. Among the available options, we selected the Google Speech API for its exceptional Speech Recognition API. This API effectively converts spoken text from a microphone into written text, also known as

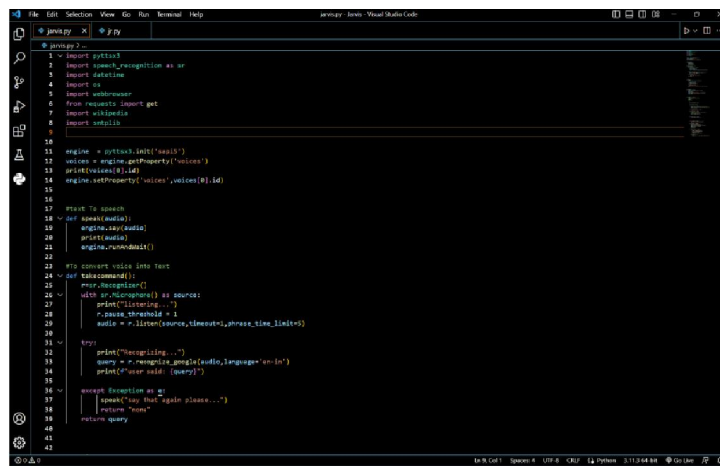
Speech to Text. By leveraging the API, spoken words are accurately transcribed into written text, particularly for the English language.

Wolfram Alpha API: The Wolfram Alpha API allows clients to submit free-form queries similar to those entered on the Wolfram|Alpha website, and the computed results are returned in various formats. This API follows the standard REST protocol, providing a straightforward means of accessing and utilizing the computational capabilities of Wolfram Alpha.

The pyttsx3 library in Python serves as a text-to-speech conversion tool. Notably, it operates offline and offers compatibility with both Python 2 and 3, making it a preferred choice over alternative libraries for text-to-speech functionality.

#### IV. RESULTS

##### 1. Opening VS-CODE

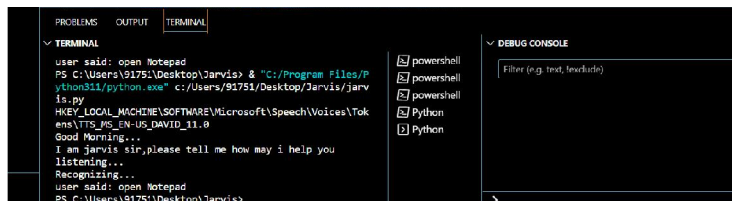


```

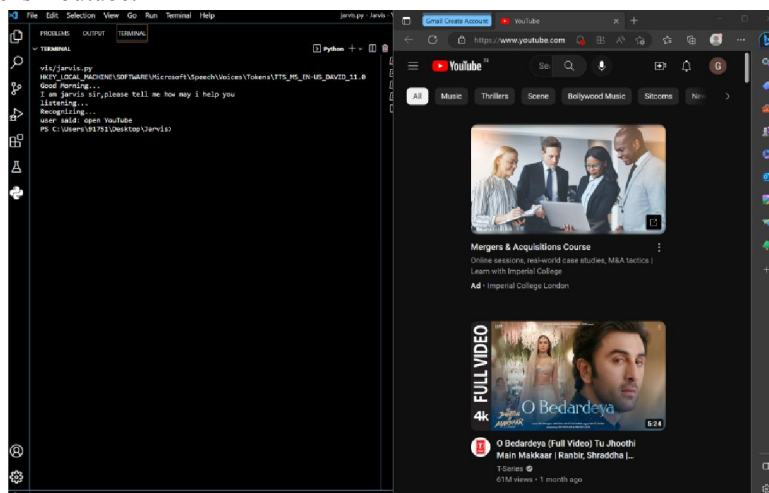
1 | import pyttsx3
2 | import speech_recognition as sr
3 | import datetime
4 | import sys
5 | import webbrowser
6 | from requests import get
7 | import wikipedia
8 | import os
9 |
10 |
11 | engine = pyttsx3.init('sapi5')
12 | voices = engine.getProperty('voices')
13 | print(voices[0].id)
14 | engine.setProperty('voices', voices[0].id)
15 |
16 |
17 | #start to speak
18 | def speak(audio):
19 |     engine.say(audio)
20 |     engine.runAndWait()
21 |
22 | #to convert text into text
23 | def takeCommand():
24 |     #using microphone
25 |     r = sr.Recognizer()
26 |     with sr.Microphone() as source:
27 |         print("Listening...")
28 |         r.pause_threshold = 1
29 |         audio = r.listen(source, timeout=5, phrase_time_limit=5)
30 |
31 |     #if
32 |     print("Recognizing...")
33 |     query = r.recognize_google(audio, language='en-in')
34 |     print(f"User said: {query}")
35 |
36 | #search knowledge on google
37 | def searchKnowledgeOnGoogle(query):
38 |     if query == "open Notepad":
39 |         speak("open Notepad")
40 |         os.system("notepad")
41 |
42 |
43 |

```

##### 2. Providing user input in terminal

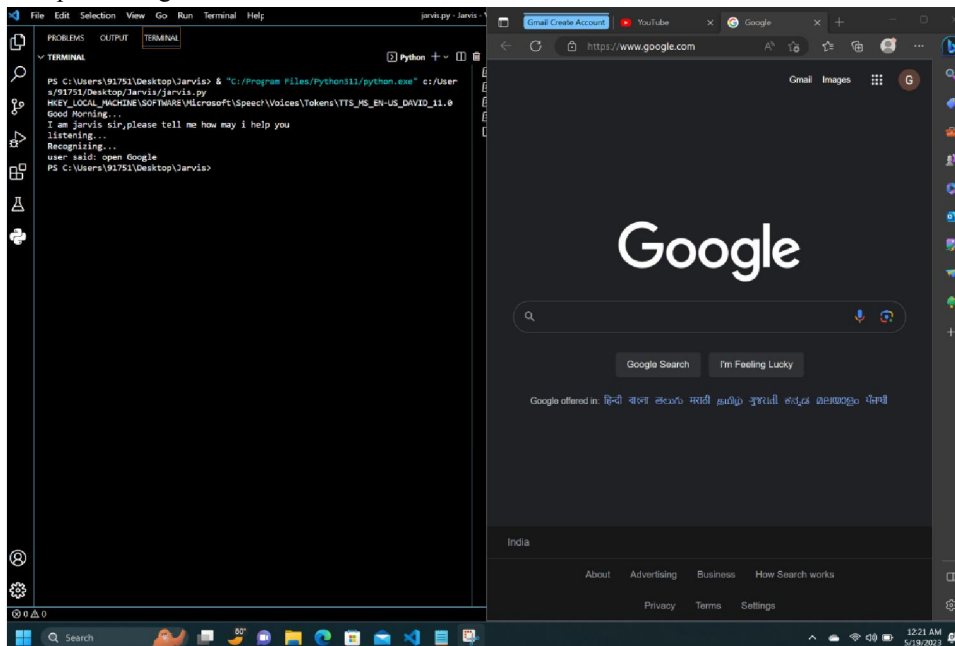


##### 3. Virtual Assistant Opens Youtube.





4.Virtual assistant opens Google



**V. CONCLUSION**

**Project Development and Implementation:**

The project's core objective revolves around the development of Windows-based software using the Python programming language, along with various APIs for Google products and mathematical operations. A team of three developers collaborated on the project, following the Incremental model and Extreme programming model. Throughout the six-month development cycle, the team adhered to a consistent process, involving requirements analysis, design, pair programming for implementation, and rigorous testing. The development process was guided by a well-defined planning strategy that determined the work approach, allocated weekly time, identified necessary resources, and provided guidance for problem-solving.

Under this development model, the project was efficiently completed, leveraging the early identification of resources that proved valuable during implementation.

**Project Usage & Potential Prospects:**

The project offers immense utility and promising prospects across various industries. While the primary focus is on creating a virtual assistant on Windows-based systems using voice recognition, the concept of speech recognition holds potential for application in diverse fields. Speech recognition technology offers convenience, time-saving benefits, and enhanced assistance for individuals facing challenges with manual operations. The program encompasses a collection of 15-20 frequently used functions on Windows PCs, providing users with a range of services within a user-friendly platform. It offers simplicity in operations compared to traditional working methods that require familiarity with desktop systems.

Moreover, the program's voice-operated feature caters to individuals who prefer voice commands and those with disabilities or difficulties in performing manual operations. The program's primary objective is to deliver services through voice interaction, making it accessible to a wider user base.

**REFERENCES**

[1]. S Subhash; Prajwal N Srivatsa; S Siddesh; A Ullas; B Santhos. Artificial Intelligence-based Voice Assistant, IEEE, 2020. <https://ieeexplore.ieee.org/document/9210344/>

- [2]. EV Polyakov, MS Mazhanov, AY Voskov, LS Kachalova, MV and SV Polyakov, "Investigation and development of the intelligent voice assistant for the IOT using machine learning", Moscow workshop on electronic technologies, 2018.
- [3]. Veton Kepuska and Gamal Bohota, "Next generation of virtual assistant (Microsoft Cortana Apple Siri Amazon Alexa and Google Home)", IEEE conference, 2018.
- [4]. Laura BURbach, Patrick Halbach, Nils Plettenberg, Johannes Nakayama, Matrina Ziefle and Andre Calero Valdez, "Ok google Hey Siri Alexa. Acceptance relevant of virtual voice assistants", International communication conference IEEE, 2019
- [5]. X. Lei, G. Tu, A. X. Liu, C. Li and T. Xie, "The insecurity of home digital voice assistants - amazon alexa as a case study", CoRR, vol. abs/1712.03327, 2017.
- [6]. J. Gratch, N. Wang, J. Gerten, E. Fast and R. Duffy, "Creating rapport with virtual agents" in Intelligent Virtual Agents, Berlin, Heidelberg:Springer Berlin Heidelberg, pp. 125-138, 2017.
- [7]. B. Weiss, I. Wechsung, C. Kühnel and S. Möller, "Evaluating embodied conversational agents in multimodal interfaces", Computational Cognitive Science, vol. 1, pp. 6, Aug 2015.
- [8]. Y. Matsuyama, A. Bhardwaj, R. Zhao, O. Romeo, S. Akoju and J. Cassell, "Socially-aware animated intelligent personal assistant agent", Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 224-227, 2016.
- [9]. M. Schroeder, E. Bevacqua, R. Cowie, F. Eyben, H. Gunes, D. Heylen, et al., "Building autonomous sensitive artificial listeners", IEEE transactions on affective computing, vol. 3, pp. 165-183, 2018.
- [10]. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, et al., "Deep speech: Scaling up end-to-end speech recognition", CoRR, vol. abs/1412.5567, 2014.
- [11]. M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research development and teaching", International Journal of Speech Technology, vol.
- [12]. P. Cosi, F. Tesser, R. Gretter, C. Avesani and M. Macon, "Festival speaks italian!" in EUROSPEECH 2001 Scandinavia 7th European Conference on Speech Communication and Technology 2nd INTERSPEECH Event, pp. 509-512, 2001.
- [13]. D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar and A. I. Rudnicky, "Pocketsphinx: A free real-time continuous speech recognition system for hand-held devices" on Acoustics Speech and Signal Processing Proceedings, May 2018
- [14]. Think Python Book, Allen Downey
- [15]. A Python Book Author: Dave Kuhlman
- [16]. Python by Narayana Book