Jarvis: Virtual Voice Command Desktop Assistant

Uttam Adha
Student
Chandigarh University
Chandigarh, India

Shayna Singh
Student
Chandigarh University
Chandigarh, India

Chanchal
Student
Chandigarh University
Chandigarh, India

Ishika Tinna
Student
Chandigarh University
Chandigarh, India

Ashish Kumar
Er., Associate Professor
Chandigarh University
Chandigarh, India

Abstract:- Earlier As everyone is aware, technology and artificial intelligence play a huge role in daily life. AI-driven voice assistants have merged with technology to become a necessary component of our everyday life. With a personal virtual assistant namely JARVIS, a user can ask questions and issue commands in the same way they would with a human. Some simple actions can even be performed with a voice command, such as opening programmers, playing music, searching Wikipedia without opening a browser, and opening apps. To improve its usability and functionality, the assistant integrates technologies including speech recognition.

Keywords:- *Voice Assistant, Voice recognition, Python Library, Jarvis AI, Chatbot.*

I. INTRODUCTION

In the present moment, machines are transforming everything that a human being can do. The performance exchange is one of the main reasons. The idea of a digital assistant consequently surfaced. An AI assistant can identify specific orders from a user and utilise that information to remove background noise and provide relevant information. Despite being entirely computer-based, AI-powered assistants have recently been incorporated into a wide range of gadgets.

Some, like Alexa, were developed especially for stand-alone devices. Given how quickly times are changing, it is more crucial than ever to teach our machines via deep learning, machine learning, and neural networks. Voice Assistant has made it possible for us to converse with our machines. These days, voice assistants are used by all major corporations to enable voice interaction between their clients and their gadgets. We're therefore moving on to the next stage of development with the Voice Assistant, enabling us to have conversations with our device.

These are some of the most crucial jobs that a voice assistant may aid you with:

- Reading the newspaper;
- Getting emails with updates
- Perform an online search;
- Watch or listen to a video.
- Putting a reminder and an alarm on
- Launch any application or tool.

These are only a few examples; we will complete a great deal more work in compliance with our specifications.

We have developed a Voice Assistant that works with Linux and Windows. Our voice assistant is a computer-based device that uses Python libraries and modules. Although this assistant is merely a basic model capable of performing all of the previously mentioned basic tasks, modern generation, while true in for better results, it still needs to be combined in some ways with gadget expertise and the internet of things (IoT). The version was created using Python modules and libraries, and it was taught using device learning.

A. Project Objective

The project aims to develop and deploy a voice-activated computer virtual assistant to boost user productivity and engagement. By smoothly incorporating voice commands into laptop operations, the aim is to provide a hands-free, intuitive experience that enables users to carry out tasks, obtain information, and control their computers far more efficiently and effortlessly.

B. Identification of problem

The primary problem here is that there aren't many voice-activated virtual assistants made just for PCs. As technology develops, the traditional laptop has transformed into a center for work, communication, and information access. Meanwhile, users find it difficult to efficiently handle a range of tasks while utilizing their laptop's capabilities. It is now feasible to identify a key issue: there are currently no voice controlled virtual assistants that can seamlessly integrate into the laptop environment and expedite user interactions.

C. Project Scope

A Jarvis virtual desktop assistant project can cover a wide range of functionality, making it an effective and adaptable instrument. The following is a list of possible features for a virtual desktop assistant that looks like Jarvis:

- NLP, or natural language processing, and voice recognition: Use leading-edge voice recognition software to decipher spoken instructions.[5]

It can be used to understand and reply to natural language inquiries.

- Automata monotonous tasks including file management, system tasks, volume settings change, and system endeavors.[2]

II. LITRATURE REVIEW

It looks at how virtual assistants have changed over time, covering everything from the advent of voice recognition technology to the difficulties presented by linguistic diversity and accents. Security lapses and data collection are two other privacy issues that are covered, along with some noteworthy cases.[2]

The necessity of tackling these past issues is emphasized in the section's conclusion in order to develop future virtual assistants and incorporate them into our daily lives.

A. Existing Solutions

The foundation for more sophisticated systems such as the Assistant from Google, Alexa, made by Amazon, and Siri by Apple was established by early virtual assistants like Clippy and Microsoft's Office Assistant. the techniques and inventions used to get around the difficulties virtual assistants face. It addresses the following crucial topics:

• Voice Recognition Technological Advancements: This article explores how these developments have led to a notable increase in virtual assistant accuracy. The ability to handle various dialects and accents posed a significant challenge for voice recognition technology. Early virtual assistants had trouble understanding a wide range of language variations. Yet, Methods such as deep learning models, and more especially the use of RNNs and convolutional neural network models (CNNs), have shown remarkable success in handling accents and dialects. [2][3]

• Natural Language Understanding: Examines how NLP models have evolved and how they can help virtual assistants comprehend context, intent, and user sentiment more fully. Sentiment analysis, context-aware algorithms, and neural network-based techniques are all examined in this.[5]

B. Goals/Objective

To advance the virtual assistant industry, it is imperative to establish well-defined goals and objectives. The main aims and objectives are described in this section. They could consist of:

- Increased Productivity: Help users be more productive by helping with scheduling and document management tasks. Jarvis can be spoken to to find particular files, folders, or documents on a user's computer or cloud storage account, and research. Provide functionalities that streamline the user's workflow by enabling the virtual assistant to effectively handle calendars, reminders, and document organizations.
- Seamless Integration: Make the virtual assistant's compatibility with desktop programs and services even better. Jarvis improves user experience by integrating with popular software, email clients, project management tools, collaboration software, cloud storage services, and more. This makes it simpler for people to handle their digital tasks and activities.
- Text and Voice Interaction: To make things easier for users, allow text- and voice-based interactions. Provide a strong natural language processing system so that users can easily interact with the virtual assistant by texting or speaking with it. This method makes sure that users can communicate with Jarvis in a way that best meets their needs at any given time by acknowledging the variety of user preferences and the various contexts in which the virtual assistant may be utilized.

III. DESIGN FLOW/PROCESS

A. Selection of Specification/Features

- Essential Features: Features that are necessary for the virtual desktop assistant's basic operation.
- Desirable Features: Not absolutely necessary improvements that would significantly improve the user experience
- Optional Features: Extra features that could be included if time and funding allow.
- User Interface (UI): Creates an intuitive graphical user interface.
- Security: To safeguard sensitive data, strong security measures are put in place.
- Voice Recognition: Voice commands can be used by users to communicate with the system.
- Task Automation: Boost productivity by automating tedious tasks.
- Natural Language Processing: Facilitates comprehension and replying to natural language inquiries by the system

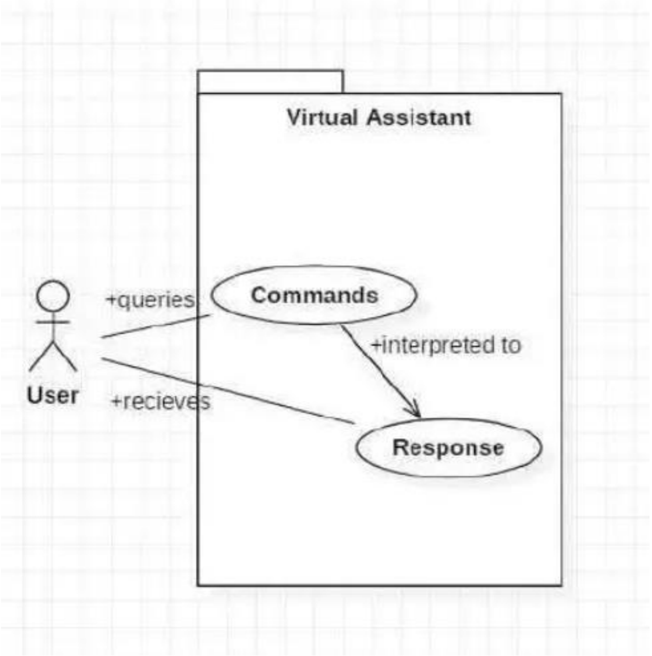


Fig 1 Use Case Diagram

IV. IMPLEMENTATION PLAN/ METHODOLOGY

The following is the implementation plan for our product-selling online auction website:

- Define requirements: Considering the virtual desktop assistant's functionality, design, and performance, specify what is required.
- Select technology stack: Decide which programming language, application, and additional system are needed for the technology stack of the application.
- Develop GUI: Use PyQT5 designer to create the GUI for the application.
- Develop Application: The Jarvis application will be made in Python.
- Test Application: Verify that the application has undergone extensive testing to ascertain its dependability, security, and user-friendliness.
- Launch Application: After the application has been launched, users can access it freely. Make sure you have a plan in place for handling any issues that may arise during launch.

You can create a reliable and useful Jarvis virtual desktop application that satisfies user needs by adhering to this implementation plan.

A. Understanding of Communication

The system uses python library for speech recognition system to convert speech to text. The verbal expression Using the microphone, user can provide tasks from the distance to the computer. The microphone then send task to cross check from queries to perform tasks in computer.[4]

B. Python Programming

The Python language takes the output from the voice recognition python module to determine if the speech or command output is an API call, a system call, or a context extraction. The output is then sent back to the Python code to provide the user with the desired results. Context extraction, or CE, is the process of removing structured data from materials that can be read by machines.

C.Context Extraction

Including this context extraction for obtaining proper data from materials that are accessible to machines is called context extraction, or CE. are either partially or fully structured. This activity usually requires analyzing texts utilizing natural language processing in human language (NLP). TEST RESULTS for context extraction are revealed by recent efforts in the processing documents in multimedia, such as automatic annotation and content extraction from images, audio, and video.

D. Calls to Systems

When computer software requests a service from the operating system's kernel while it is running, this is known as a system call. Examples of this opening files folder or any file from any location, opening applications like notepad or word in which we can write or delete the text, and getting information about system or battery percentage. System calls serve as the conduit between a task and the computer's operating system.

E. Text to speech

Machines that possess TTS (text-to-speech) capability can read content loudly. Written text is converted into alphabetical transcription, which is then converted into waves that can be produced as sound by a TTS Engine. [4]

V. RESULT

Natural language processing, or NLP, is a technique utilized by speech recognition software to break down content into comprehensible chunks, transform them into computer-readable forms, and then analyze the individual components. The software then makes decisions based on speech patterns and programming. After figuring out what the users said, it turns the conversation into text, which is then saved in a variable. Because our voice assistance is text-dependent, the command is sent in the form of a text, which looks up similar text written as functions. If the text matches, the function is performed; if not, an exception is addressed.

A. GUI Design:

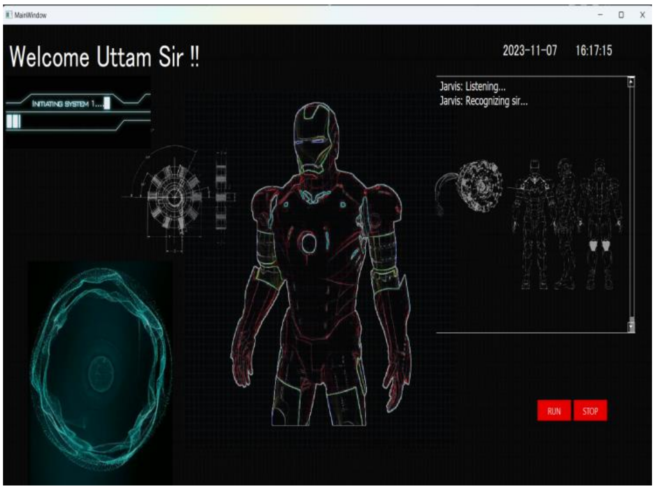
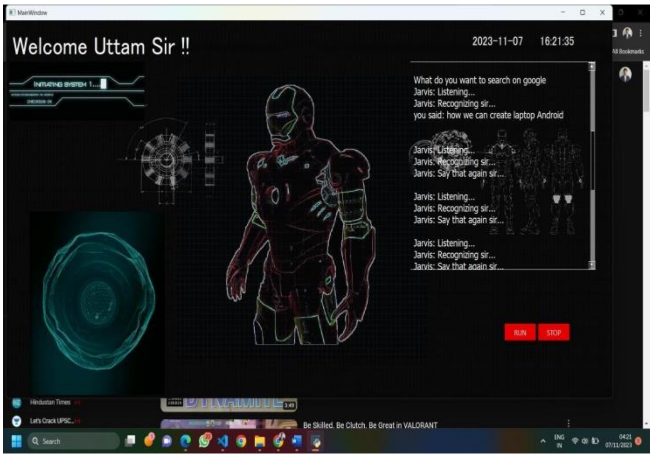

Welcome Uttam Sir !!
2023-11-07 16:17:15
Jarvis: Listening...
Jarvis: Recognizing sir...

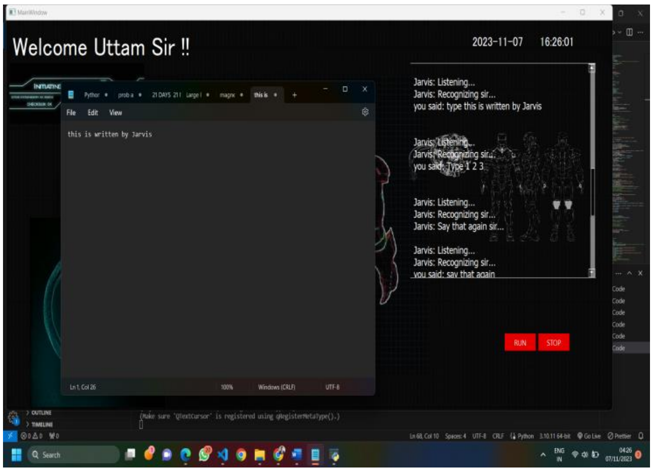
Fig 2 Home Page

This is the front-end portion of the project's GUI, which consists of some background GIFs, Run and Stop button and one output box using PyQt5. In this context, A complete set of Python bindings for Qt v5 is called PyQt5. It can be used as an alternative application development language to Python and is implemented as more than 35 extension modules.



```
What do you want to search on google
Jarvis: Listening...
Jarvis: Recognizing sir...
you said: how we can create laptop Android

Jarvis: Listening...
Jarvis: Recognizing sir...
Jarvis: Say that again sir...

Jarvis: Listening...
Jarvis: Recognizing sir...
Jarvis: Say that again sir...

Jarvis: Listening...
Jarvis: Recognizing sir...
Jarvis: Say that again sir...
```

Fig 3 Home Page [Command and Response]



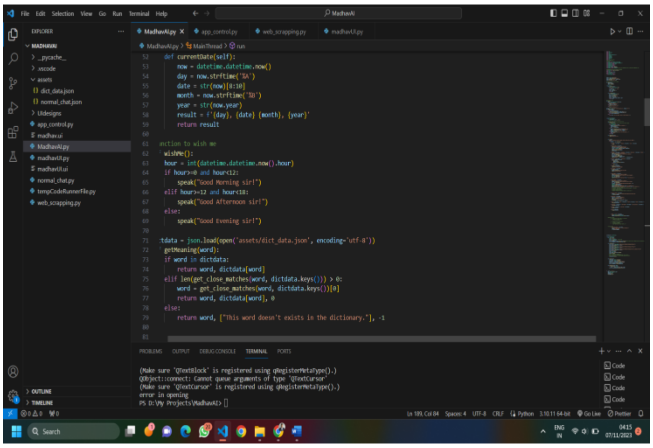
```
this is written by Jarvis
```

```
Jarvis: Listening...
Jarvis: Recognizing sir...
you said: type this is written by Jarvis

Jarvis: Listening...
Jarvis: Recognizing sir...
you said: Type 1 2 3

Jarvis: Listening...
Jarvis: Recognizing sir...
Jarvis: Say that again sir...

Jarvis: Listening...
Jarvis: Recognizing sir...
you said: say that again
```

```
(Make sure 'QTextCursor' is registered using qRegisterMetaType().)
```

Fig 4 Home Page [Response]

B. Code Implementation



```python
    def currentDate(self):
        now = datetime.datetime.now()
        day = now.strftime('%A')
        date = str(now)[8:10]
        month = now.strftime('%B')
        year = str(now.year)
        result = f'{day}, {date} {month}, {year}'
        return result

unction to wish me
 wishMe():
  hour = int(datetime.datetime.now().hour)
  if hour>=0 and hour<12:
      speak("Good Morning sir!")
  elif hour>=12 and hour<18:
      speak("Good Afternoon sir!")
  else:
      speak("Good Evening sir!")

:tdata = json.load(open('assets/dict_data.json', encoding='utf-8'))
 getMeaning(word):
  if word in dictdata:
      return word, dictdata[word]
  elif len(get_close_matches(word, dictdata.keys())) > 0:
      word = get_close_matches(word, dictdata.keys())[0]
      return word, dictdata[word], 0
  else:
      return word, ["This word doesn't exists in the dictionary."], -1
```

```
(Make sure 'QTextBlock' is registered using qRegisterMetaType().)
QObject::connect: Cannot queue arguments of type 'QTextCursor'
(Make sure 'QTextCursor' is registered using qRegisterMetaType().)
error in opening
PS D:\My Projects\MadhawAI> 
```

Fig 5 Code using Python language.

VI. CONCLUSION AND FUTURE WORK

In this paper, we covered the use of Python for Jarvis's voice commands for Windows. Jarvis's conversation aid has made life easier for humans. For this project, we're using computational intelligence technology, so Jarvis voice assistance works with all Windows versions. Jarvis's speaking assistance can do all the duties of other assistance, including some special ones, just like the search engine, screenshots, application open, typing, windows switch, locking, sleeping, and shutting down devices with our voice input. We know that Jarvis' vocal help will last forever.

Future work will entail creating a feature that facilitates making payments when purchasing online and ensuring that voice assistance functions on all Python versions running on Linux and Windows similar.

ACKNOWLEDGEMENT

We want to sincerely thank our supervisor, Er. Ashish Kumar, an associate professor at Chandigarh University in India, for all of her hard work and dedication in seeing this project through to completion. We sincerely appreciate all of your advice and help as we completed the project.

REFERENCES

[1]. R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, "JARVIS: An interpretation of AIML with integration of gTTS and Python," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 486-489, doi: 10.1109/ICICICT46008.2019.8993344.

[2]. P. Dalal, T. Sharma, Y. Garg, P. Gambhir and Y. Khandelwal, "“JARVIS” - AI Voice Assistant," 2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP), BHOPAL, India, 2023, pp. 273-280, doi: 10.1109/IHCSP56702.2023.10127134.

[3]. Yang, D., Zhou, M., and Huang, J., January 2007. Knowledge Extraction from Online Discussion Forums for Chatbots. IJCAI, Volume 7, pages 423–428.

[4]. M. Gupta, R. Kumar and H. Sardalia, "Voice Assistant Technology: The Case of Jarvis AI," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-5, doi: 10.1109/INCET57972.2023.10170362.

[5]. L.K. Fryer and R. Carpenter. (2006). Bots as tools for learning languages. Tech and Language Learning.