**A**

**Synopsis**

**on**

**DESKTOP ASSISTANT USING PYTHON**

**Submitted by**

**Aditya Agarwal**

**2100290140008**



**Submitted to:**

**Mr. Ankit Verma Sir**

**Department of Computer Applications,**

**KIET Group of Institutions,**

**Delhi-NCR, Ghaziabad**

**September 2022**

**ABSTRACT**

This project is about how new technologies can be used to develop an intelligent Virtual Desktop Assistant that focuses on user-based data. It will analyze the possible utility of one single piece of software as a Virtual Desktop Assistant by looking at examples of intelligent programs with natural language processing that are now available, with various categories of support.

Natural Language Processing is used to activate the ability to communicate socially, storing (and evaluating) information in the context of the user. New technology, it is suggested, may soon make the concept of virtual assistants a reality. Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with natural language processing algorithms in the form of a Virtual Desktop Assistant, with basic natural language processing and the ability to function without the need for another type of human input (or programming) may already be viable.

**INTRODUCTION**

Human interaction is rapidly being supplanted. Performance is one of the key reasons behind this shift. Rather than progress, technology has undergone a significant transformation. In today's world, we use technologies like Machine Learning and Neural Networks to teach our machines to do their jobs on their own or to think like people. With the help of virtual assistants, we may now communicate with our machines in the modern world. Companies such as Google, Apple, Microsoft, and others have virtual assistants such as Google Now, Siri, and Alexa that allow users to operate their machines just by speaking to them. These types of virtual assistants are beneficial to the elderly, the visually and physically challenged, children, and others by ensuring that interacting with machines is no longer a challenge. Even blind persons who are unable to see the computer can communicate with it just through their voice. The following are some of the basic tasks that most virtual assistants can help with:

• Aggregate Percentage Calculator

• Search on Wikipedia

• Open Any Websites

• Random Password Generator

Our voice assistant is a desktop-based application created with Python modules and libraries. This assistant is only a basic version that can do all the duties listed above. All you have to do is give the assistant a command, and the assistant will take care of the rest. There will be no need to write extensive codes to execute a task with the help of voice-activated virtual assistants. The system will do it for us. It will let you tell the time, day, and temperature, and answer to many more commonly asked questions.

The smart functions which can be performed by the desktop assistant will be like opening new tabs playing music, and videos, drafting messages, scheduling reminders, and opening system program applications like MS Word, MS Excel, Notepad, and many more applications. It will let you capture a screenshot by just the voice command and let you save them by a special notation by which you can remember when it was taken and what for was it taken.

**TECHNOLOGY USED**

1. **Python**

Python is a high-level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in the easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, Data science, etc. Python has a lot of libraries for every need of this project. For JARVIS, libraries used are speech recognition to recognize the voice, Pyttsx3 for text-to-speech, selenium for web automation, etc. Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most of the time and implement just that part more efficiently in some lower-level languages. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

1. **AI**

Before 1949, computers could execute commands, but they could not remember what they did as they were not able to store these commands. In 1950, Alan Turing discussed how to build intelligent machines and test this intelligence in his paper “[Computing Machinery and Intelligence](https://www.csee.umbc.edu/courses/471/papers/turing.pdf).” Five years later, the first [AI program](https://history-computer.com/inventions/logic-theorist-complete-history-of-the-logic-theorist-program/) was presented at the [Dartmouth Summer Research Project on Artificial Intelligence (DSPRAI)](https://aaai.org/ojs/index.php/aimagazine/article/view/1911/1809). This event catalyzed AI research for the next few decades.

Computers became faster, cheaper, and more accessible between 1957 and 1974. Machine learning algorithms improved and, in 1970, one of the hosts of DSPRAI told Life Magazine that there would be a machine with the general intelligence of an average human being in three to eight years. Despite their success, computers’ inability to efficiently store or quickly process information created obstacles in the pursuit of artificial intelligence for the next ten years.AI will help in recognizing speech and let the machine think what the user wants to do and wants the machine to do and will let the machine do accordingly to that.

**HARDWARE AND SOFTWARE REQUIREMENTS**

* Processor i5 and above
* 6 GB Ram and above
* Windows 8 and above

**MODULES**

1. **Speech Recognition**

This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition, etc.

1. **Pyttsx3**

Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3. x versions. Its main advantage is that it works offline

1. **Wikipedia**

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the Media Wiki API so you can focus on using Wikipedia data, not getting it.

1. Date-Time

This class contains information on both dates and times. Like a date object, Date Time assumes the current Gregorian calendar is extended.

**FUTURE SCOPE**

The virtual assistants which are currently available are fast and responsive, but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios.

The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc., and IoT. With the incorporation of these technologies, we will be able to achieve new heights. What virtual assistants can achieve is much beyond what we have achieved till now. Most of us have seen Jarvis, that is a virtual assistant developed by an iron man that although fictional, has set new standards of what we can achieve using voice-activated virtual assistants. Our assistant will decrease the gap between computers and old age people, and it will also help the disabled, handicapped, and disabled people to perform tasks very easily. After using this people doesn’t have to depend on others for their task if they don’t know how to do them.

**Functioning of the Project**

* The user has to give access after installation to let the assistant access the path for opening programs.
* Users don’t need to register they can perform tasks by just installing it.
* Users need to speak the thing which they want the assistant to do.
* Tasks that can be performed by the assistant will be speaking Day, Time, Temperature, taking screenshots opening system applications typing messages by speaking, and playing music.
* The user needs to speak properly as their voice will be taken as the command.

**CONCLUSION**

The desktop assistant system will use natural language processing and can be integrated with artificial intelligence techniques to achieve a smart assistant that can control the computer and applications and even solve user queries using web searches.

It can be designed to minimize the human efforts to interact with many other subsystems, which would otherwise have to be performed manually by achieving this the system will make human life comfortable.

**GANTT CHART**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| – | **WEEK**  **1** | **WEEK 2** | **WEEK 3** | **WEEK 4** | **WEEK 5** | **WEEK 6** | **WEEK 7** | **WEEK 8** | **WEEK 9** | **WEEK 10** | **WEEK 11** | **WEEK 12** |
| Requirement analysis and feasibility check |  |  |  |  |  |  |  |  |  |  |  |  |
| Designing |  |  |  |  |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing and maintenance |  |  |  |  |  |  |  |  |  |  |  |  |