

# Enhancing Desktop Productivity: An Approach of Virtual Assistance for Desktop Workflow

Satyam Singh

School of Computer Science and Engineering

Lovely Professional University

satyamr1109@gmail.com

Aryan Mahale

School of Computer Science and Engineering

Lovely Professional University

aryan.mahale@outlook.com

Ishan Kumar

School of Computer Science and Engineering

Lovely Professional University

ishan.23347@lpu.co.in

## Abstract

*Assistant technologies such as Google Assistant, Alexa, and Siri have attracted much attention because of their potential to revolutionize human-computer interaction. In this age where time is the most important commodity, no one wants to waste their time writing down their requests on a computer, so assistant technologies come into play, allowing people to complete their tasks via voice command instead of typing their requests. The problem, however, is that this technology is only suitable for cell phones and browsers for Internet searches. Therefore, we have developed a system that supports users in any type of task that can be performed on the desktop, such as searching on the Internet, controlling the system, opening files and software, and much more.*

## I. INTRODUCTION

As time passes technology also advances and we are enhancing the existing technology for maximum use and making life simple as it can be. Already many voice assistants exist but desktop assistant has their own importance especially in

case of Windows. The project is similar to a voice assistant which takes voice input command and performs the task requested by the user. Typing the queries is a boring and time-consuming process, the solution to this big obsolete process of typing is that we switch over to an assistant which understands us and does the initial work for us. An assistant is the best replacement for typing commands.

Every software has as its top goal protecting the device's security and the private information it stores and processes. Desktop assistance is made in both ways, and we set some security features as well to make it secure like not everyone can use them without having permission.

## II. LITERATURE REVIEW

The history of speech recognition technology can be traced back to the early 1950s, when Bell Laboratories developed the first speech recognition system, called the "Audrey" system. This system could only recognize numbers spoken by a single person then by 1980s, Researchers began using hidden Markov models (HMMs) to

improve speech recognition technology. These models were capable to analyse sound patterns in human speech and enable the recognition of different words and phrases. However, the technology was still limited by the capabilities of the hardware, and then came the time of Commercialization of speech recognition. In 1990s, it began to be used in several commercial applications, such as telephone customer service and dictation software for computers. However, these early systems were still limited in their accuracy and could only recognize limited range of speech. This continued until 2000s when Google launched the Google Voice Search app (now known as Google Assistant) with features such as: Expanded vocabulary, adaptation to a noisy environment, improved accuracy, real-time processing, and most importantly, integration with other technologies Google Voice Search revolutionized speech recognition by introducing a new approach that combined advanced machine learning techniques with the processing power of Google's servers. The technology got better every year, and the beginning of the 2011 decade saw an explosion of other voice recognition apps with more features and greater efficiency and accuracy. In 2011, Apple introduced Siri as a feature of the iPhone 4s that resembled Google's voice search. With Amazon's Alexa and Google Home having advanced features like smart home integration, third-party app integration, multilingual support, we've seen consumers become more comfortable talking to machines.

Overall, the history of speech recognition technology is characterized by a steady evolution and improvement in accuracy and performance. While early systems were limited to single word recognition, modern systems can understand natural language and perform complex tasks, making speech recognition a powerful tool for a variety of applications.

### III. METHODOLOGY

The whole project is divided into two parts, the first is for validation, the second is the desktop assistant.

1. In order to fully use a software or a virtual assistant, everyone prefers security features. For example, Siri and our desktop assistant are also safe from unauthorised users. To use the desktop assistant, users must confirm their identity via a manual login. There is a separate UI for both registration and login. If the user is new and wants to explore the desktop assistant, he must first register.

2. The second part is about how the Desktop Assistant works. The Desktop Assistant accepts voice commands as input, recognises the command and performs the operation requested by the user. If the command cannot be recognised, the input is repeated. For example, if the user is asking the assistant about the current live weather details, the assistant will respond with voice command the exact current weather report of the user like current temperature, humidity wind speed etc.

#### **System Architecture:**

The program presents a main page to the user, giving the option to either login or register as a new user. New users must fill out all required details and submit them for storage in a MySQL database. Registered users can log in using their unique number and password to activate the Desktop Assistant, which allows them to perform various tasks like Google and YouTube searches, access live location information and weather details, and more. If a user enters invalid login credentials, a popup will appear prompting them to try again.

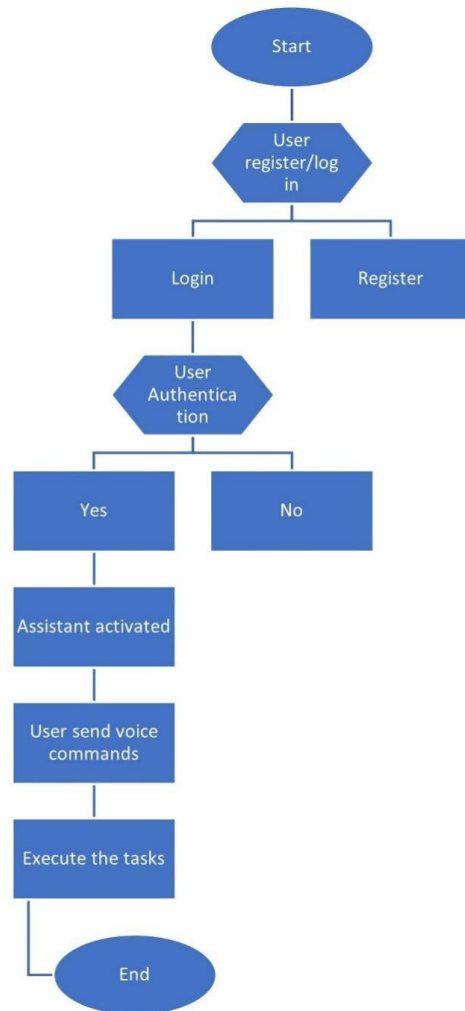


Figure 1: System Architecture Flow chart.

### Libraries used:

**Tkinter:** tkinter is a Python library used to create the GUI of any project. It provides a variety of options or widgets such as labels, buttons, text boxes and many others.

**mysql.connector:** For storing and retrieving user login data, a MySQL database is utilized. This is achieved using the mysql.connector library, which connects Python to the MySQL database, allowing for the manipulation of data in the database. It is used in the project to add new data to the database when a new user registers and then retrieve this data to validate a user during login.

**pyttsx3:** Pyttsx3 is a Python library used for converting text to speech. It provides an easy way to convert written text to spoken words with a variety of voices.

**speech\_recognition:** The speech recognition module in Python can recognize spoken words from an audio source (e.g., a microphone or audio file) and convert them to text for further use in other operations.

**datetime:** The Python datetime module provides classes for working with dates and times, allowing programmers to perform tasks such as formatting dates and performing arithmetic operations. We can use this module to get the current time and date.

**pywhatkit:** It helps in automating tasks like sending emails, WhatsApp messages and performing web searches. In this project, this module is used to perform YouTube and Google searches efficiently.

**Geocoder:** This library allows performing geocoding tasks like finding coordinates of an address or nearest address to given coordinates. In the project, it is used to find the current latitude and longitude of a user.

**Wikipedia:** It is used to easily access and extract information from Wikipedia, the largest online encyclopedia in the world. It provides an easy way to programmatically search, retrieve, and extract information from Wikipedia articles.

**webbrowser:** The webbrowser module in Python simplifies opening and viewing web pages in the user's default browser.

**os:** It provides various functions for managing files, directories, processes and the file system.

**pyautogui:** This module allows you to automate mouse movements, keyboard input and other GUI interactions in your Python program.

**requests:** Simplifies interaction with web services and APIs. In the project it is used to retrieve the live weather information via the API ("openweathermap").

**Googletrans:** It allows you to easily use Google Translate for text translation in different languages in your Python application. We call translate () method on the translator object and pass in the text to translate as well as the destination language.

## IV. RESULT

The results of complete project can be divided into 3 main categories:

1. G.U.I. of the project, which includes main page, signup and login page.
2. Second category includes the MySQL database, that is used to store the user information.
3. If the login is successful than comes the third and the main category i.e., "The Desktop Assistant".

As soon as the Assistant is initiated it will start Listening and recognising the command of the user. If the command given by the user is recognised by the assistant, it will start executing the operation requested by the user.

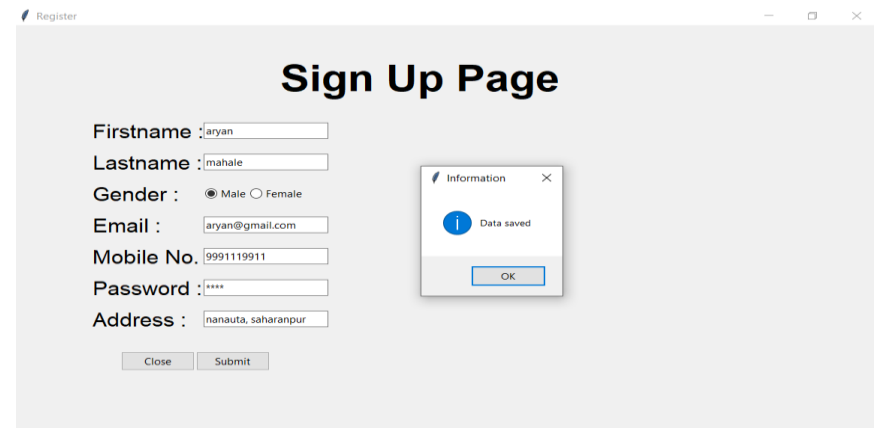


Figure 2: Register page for registering a new user.

```
mysql> select * from member;
+-----+-----+-----+-----+-----+-----+-----+
| fname | lname | gender | email          | mobile | passwd | address      |
+-----+-----+-----+-----+-----+-----+-----+
| satyam | singh | 1      | satyam@gmail.com | 9548149557 | abcd  | chaura      |
| Shivam | Singh | 1      | shivam@gmail.com | 9991119949 | dcba  | nanauta     |
| aryan  | mahale | 1      | aryan@gmail.com  | 9991119911 | dcba  | nanauta, saharanpur |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Figure 3: Data entered on registering a new user is saved in MySQL database.

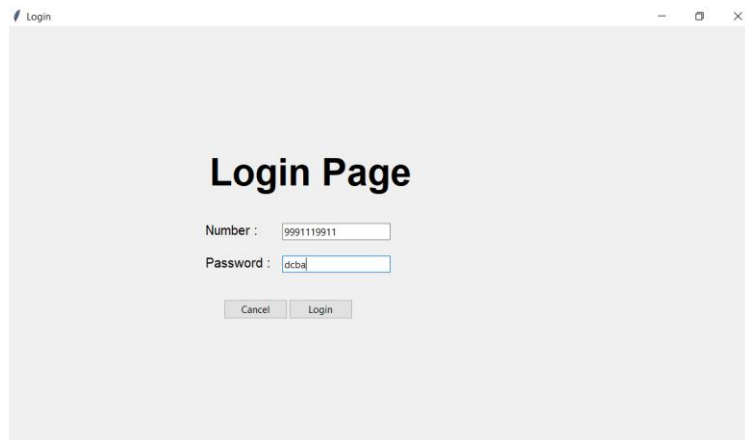


Figure 4: Login page with user number and password.

```

find : if ucy
you said how is the weather today
todays temperature is 31.93 degree selsius
with , Haze
and , Humidity35 percent
and , wind speed2.06meter per second

```

Figure 5: system showing the weather details.

```

you said what is my current location
sir, your current location is
Ludhiana
Punjab
with latitude
30.912
and longitude
75.8538

```

Figure 6: System showing the result for current location of the user.

## Main functionalities

System is capable to execute the commands asked by the user and give the desired results. User can ask the system to perform lots of operations like –

1. Simple conversation
2. Language translation
3. Live Weather details for the current location of the user:
4. Giving Current location of the user
5. Increasing and decreasing the Volume and brightness as well
6. Opening any software like Notepad, PowerPoint, Vs Code, Dev C++
7. Opening Google and YouTube
8. Searching on Wikipedia and giving voice output
9. Searching on google and giving voice output for user's query.
10. Searching user specified video on YouTube.
11. Opening Websites like Instagram, Flipkart.
12. Specifying current date
13. Specifying the current time
14. Opening the user asked directory like downloads, Desktop
15. Playing movies and music that are available in the system.
16. Capturing photo on user's voice command
17. Telling jokes
18. Shutting down the system

## V. CONCLUSION

The main goal of the project was to develop a secure desktop assistant that first authenticates the user ID with the data available in the MySQL database and then only allows the authenticated users to submit their voice command, which then recognises the command and performs the operation requested by the user.

We also did a background search for relevant available desktop assistant but did not find any desktop assistant with security features and all possible functions that can be performed on Windows. So, we have built one with the security feature and all the possible operations that can be performed on the Desktop like – Live Weather information, Current Location, Language Translation, Google and YouTube Search, Increasing/Decreasing the brightness and volume of the system and much more as explained above. We have built the complete system using the Python language. The availability of open-source libraries and resources has made it easier for us to create custom wizards that meet specific requirements, and the code was implemented using Vs Code integrated development environment IDE.

## REFERENCES

- [1]. Haris Khan, “Iron Man Jarvis AI Desktop Voice Assistant “,  
<https://www.codewithharry.com/videos/python-tutorials-for-absolute-beginners-120/>
- [2]. <https://pypi.org/project/pyttsx3/>
- [3]. Gaurav Kumar, “Voice Assistant Using Python”,  
<https://www.geeksforgeeks.org/voice-assistant-using-python/>

- [4]. How to use API with Python (Beginner’s Guide) ,  
<https://rapidapi.com/blog/how-to-use-an-api-with-python/>

- [5]. “My Voice Assistant Using Python”,  
[https://www.academia.edu/78490692/My\\_Voice\\_Assistant\\_Using\\_Python](https://www.academia.edu/78490692/My_Voice_Assistant_Using_Python)

- [6]. “Google Assistant”, “Apple Siri”, “Amazon Alexa”.

- [7]. “A brief history of voice recognition technology”  
<https://www.totalvoicetech.com/a-brief-history-of-voice-recognition-technology/>