

Voice Desktop Assistant

PROJECT REPORT

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY

By

Hriday Jain-56

Garv Jain-55

Tanisha Dube-35

Alphonz Velacherry-05

Guide

Prof. Anjali Malviya



Department of Information Technology

Thadomal Shahani Engineering College

Adv. Nari Gursahani Marg, TPS III, Off Linking Rd, Bandra West,

Mumbai, Maharashtra 400050



University of Mumbai

(Academic Year 2023-24)

CERTIFICATE

This is to certify that the Mini Project entitled “**Voice Desktop Assistant**” is a bonafide work of **Hriday Jain, Tanisha Dube, Garv Jain, Alphonz George Velacherry** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “**Bachelor of Engineering**” in “**Information Technology Engineering**”.

Prof. Anjali Malviya
Project Guide

Prof. Mukesh Israni
Head of Department

Prof. G.T. Thampi
Principal

MINI PROJECT APPROVAL

This Mini Project entitled “**Voice Desktop Assistant**” by **Hriday Jain, Tanisha Dube, Garv Jain, Alphonz George Velacherry** is approved for the degree of **Bachelor of Engineering in Information Technology Engineering**.

Examiners

1.....

Prof. Anjali Malviya

2.....

(External Examiner name & Sign)

Date:

Place: Mumbai

ACKNOWLEDGEMENT

This project has consumed a significant amount of time and resources, and we would like to thank everyone who has contributed to its success. We would like to express our deepest gratitude to the project head, **Prof. Anjali Malviya** for her time, patience and direction, as well as for inspiring us throughout. Without their guidance and support, this project would not have been possible. Their expertise, insights, and encouragement were invaluable in helping us navigate the various challenges and complexities of this project.

We would also like to extend our sincere thanks to the many research papers that we consulted throughout the course of this project. The insights and ideas presented in these papers were instrumental in helping us develop a deeper understanding of the subject matter and in shaping the direction of our research.

We are also grateful to our colleagues and friends who provided us with valuable feedback and support throughout the project. Their encouragement and insights were invaluable in helping us refine our ideas and in pushing us to achieve our best work.

1.Hriday Jain-_____

2.Garv Jain-_____

3.Tanisha Dube-_____

4.Alphonz Velacherry-_____

ABSTRACT

"Voice Desktop Assistant" is a Python-based application revolutionizing user interactions with computer systems through intuitive voice commands. This innovative tool offers a wide range of features aimed at enhancing productivity and workflow on the desktop. Its voice-activated command system enables users to perform tasks such as web searches and application launching effortlessly, eliminating the need for manual input and streamlining workflows. Additionally, the assistant supports multimedia playback from popular platforms like YouTube and Spotify, providing users with hands-free access to their favourite content. Personalized responses from the assistant further enhances user engagement, adapting to individual preferences and requirements to create a dynamic interaction experience. Robust time management tools, including setting and managing to do lists, help users stay organized and productive. It also provides a streamlined process to navigate through files and folders with the help of voice activated commands. Leveraging Google search capabilities, the assistant extracts relevant information and provides timely updates, enhancing its utility and usefulness in various contexts. The Voice Desktop Assistant represents a significant advancement in desktop computing, offering a user-friendly solution for optimizing productivity and workflow through intuitive voice commands, multimedia playback, personalized interactions, and efficient time management tools.

TABLE OF CONTENTS

Chapter 1		
1.1	Introduction	4
1.2	Motivation	
1.3	Problem Statement	
1.4	Objectives	
1.5	Organization Of Report	5
Chapter 2		
2.1	Survey of Existing Solutions	6
2.2	Limitation Existing system or research gap	7
2.3	Mini Project Contribution	
Chapter 3		
3.1	Proposed System	9
3.2	Architecture and Framework	
3.2.1	Streamlit Library	
3.2.2	Tkinter Library	10
3.2.3	Python	
3.2.4	SQL	11
3.3	User Flow	
3.4	Database Structure	12
3.5	Results	13
Chapter 4		
4.1	Future Scope	15
4.2	Conclusion	
	References	16

LIST OF FIGURES

Fig No.	Title	Page No.
1	Example of database structure for To-do list	12
2	User interface of the application	13
3	Example of functionality: News Update	13
4	Tkinter elements: Email Function	14
5	Tkinter elements: Mini Assistant	14

CHAPTER 1

1.1 Introduction

Welcome to the project review session for our Python-based desktop voice assistant, "Jarvis." In this session, we'll provide an overview of our simple yet effective project aimed at bringing basic voice-controlled functionalities to desktop environments. Jarvis represents our humble attempt at exploring the realm of voice-enabled assistants using Python. While it may not boast the complexity of commercial solutions, our project aims to demonstrate the feasibility and practicality of integrating voice commands into desktop interactions using Python.

1.2 Motivation

The idea for this project came to mind while searching for software solutions that can be helpful for physically challenged persons and people with reading and writing disabilities. We realized that there were very few, mostly expensive solutions and realized that this could be a good avenue to work on. This led to us exploring different systems to enable voice activation to streamline desktop interactions.

1.3 Problem statement

The goal of this project is to build a Python-based application that can carry out some basic functionalities of a desktop using voice commands thus leading to increased productivity.

1.4 Objectives

To tackle this challenge, our objective is to develop "JARVIS", a desktop voice assistant application which can open basic applications, open basic websites, navigate through files and folders, integrate features like sending WhatsApp messages and emails, provide news updates and any other relevant features.

By achieving these objectives, "Jarvis" aspires to be a start of a system that revolutionizes how we interact with our computers.

1.5 Organization of the report

This report is divided into three chapters. The first chapter covers the project's introduction, problem description, motivation for the topic, and objectives. The second chapter is a review of the literature. It contains all of the research effort done on this topic. This chapter covers all you need to know about studying current systems and learning new technologies. The third chapter describes the suggested system that will be employed in this project. This chapter contains screenshots of the project's block diagram, methodologies, hardware, and software. All materials referred and utilized in the development of this project are included in the references section.

CHAPTER 2

2.1 Survey of existing systems

Apple's VoiceOver feature for Mac is a powerful accessibility tool designed to assist users who may have difficulty typing or navigating a traditional graphical user interface due to motor impairments or other disabilities. VoiceOver provides a spoken description of items on the screen, allowing users to interact with their Mac using only their voice.

For individuals with typing issues, VoiceOver offers a lifeline by enabling them to control their Mac entirely through spoken commands. Whether it's navigating menus, selecting options, or even typing text, VoiceOver provides audible feedback and guidance every step of the way.

Another existing system is Amazon's Alexa, which serves as a popular voice-controlled virtual assistant integrated into various Amazon devices, smart speakers, and third-party products. Alexa offers a wide range of functionalities, including setting reminders, playing music, providing weather updates, controlling smart home devices, and even ordering products online. With its natural language processing capabilities, Alexa understands and executes commands spoken by users, providing a seamless and hands-free experience.

Google Assistant is another prominent example of a voice-controlled assistant, available on smartphones, smart speakers, and other devices. It offers similar functionalities to Alexa, such as setting reminders, answering questions, playing music, and controlling smart home devices. Google Assistant stands out with its integration with other Google services like Gmail, Calendar, and Maps, providing users with personalized assistance based on their preferences and activities. Its ability to understand context and provide relevant responses contributes to a smooth and intuitive user experience.

2.2 Limitations of existing systems or research gap

While Apple's VoiceOver feature on Mac provides invaluable support for users with typing issues, it's important to acknowledge that Mac computers may not be accessible to everyone due to factors such as cost and availability. Despite its robust accessibility features, including VoiceOver, Macs can be out of reach for individuals facing financial constraints or those living in regions where Apple products are not readily available or affordable.

Additionally, while VoiceOver on Mac offers comprehensive support for users with motor impairments or disabilities, it may not address all accessibility needs. For example, some users may require specific customization options or additional features tailored to their unique requirements, which may not be fully accommodated by VoiceOver. This limitation highlights the importance of continuously exploring and developing innovative accessibility solutions to cater to diverse user needs and preferences.

Furthermore, the reliance on voice-controlled assistants like Alexa and Google Assistant introduces potential privacy and security concerns. As these assistants constantly listen for wake words and commands, there is a risk of recording and storage of sensitive information. Users may feel hesitant to fully utilize voice-controlled assistants due to apprehensions about data privacy and the potential misuse of their personal information. Addressing these concerns and implementing robust privacy measures is essential to build trust and confidence among users, fostering broader acceptance and adoption of voice-controlled technologies.

2.3 Mini project contribution

Our project Jarvis represents more than just a technological innovation; it embodies a commitment to inclusivity and accessibility. By providing a free alternative for individuals who may face barriers to accessing similar commercial solutions, we aim to democratize the benefits of voice assistant technology. This democratization extends

beyond mere convenience; it has the potential to empower marginalized communities, including those with disabilities or limited financial resources, by granting them equal access to tools that can enhance their productivity and quality of life.

This project also allows for and creates a system where the privacy and security of the user isn't compromised. Voice interpretation is done offline without the need to connect to the internet and share voice data allowing systems to store it and use it for nefarious purposes. By ensuring that all the data given by the user stays within the control of the user itself and allowing the user to clear the user history at any point of time, it is also a significant step forward in security and privacy.

The project's ultimate goal is to make life easier for individuals. Through the seamless integration of voice assistant technology into daily routines, we seek to streamline tasks, simplify processes that present alternatives can't provide. Whether it's aiding in task management, providing access to information, or facilitating communication, JARVIS strives to be a reliable and indispensable companion, empowering users.

CHAPTER 3

3.1 Proposed system

The system will take spoken input from the user and interpret it to convert it into text form which can then be iterated through a code that searches for keywords in the converted text. Once the relevant keywords are identified, the respective actions are triggered within the code. Some examples of the following are given below:

- **Google Search:** Users can trigger the keyword “Search” followed by what they want to search.
- **To-do List:** Users can direct the assistant to edit the To-do list to store tasks that they’d like to carry out and then delete them as per their convenience.
- **Whatsapp Message:** Users can direct the assistant to send WhatsApp messages, then choose who they want to send it to and finally the contents of the message they want to send.
- **File Navigation:** To navigate to the files within the computer the user can tell the assistant to open files and then navigate through the file system.

3.2 Architecture / Framework

3.2.1 Streamlit Library:

Streamlite is a powerful Python library designed for creating interactive web applications with minimal effort. With its intuitive and straightforward syntax, developers can quickly build data-driven applications by simply writing Python scripts. Streamlit eliminates the need for complex web development frameworks and boilerplate code, allowing users to focus on their data analysis and visualization tasks. Its extensive collection of built-in widgets makes it easy to create interactive elements such as sliders, buttons, and dropdowns. Streamlit streamlines the development process, making it accessible to both beginners and experienced developers alike.

3.2.2 Tkinter Library:

Tkinter is a versatile and widely-used library in Python for creating graphical user interfaces (GUIs). It provides a simple and intuitive way to design and build desktop applications with graphical components such as buttons, labels, textboxes, and more. Tkinter's strength lies in its integration with the Tk GUI toolkit, which is cross-platform and comes pre-installed with Python. This makes Tkinter readily available for developers without requiring additional installations or dependencies. Despite its simplicity, Tkinter offers extensive customization options, allowing developers to create visually appealing and functional interfaces for their applications. With its ease of use and robust functionality, Tkinter remains a popular choice for developing desktop applications in Python.

3.2.3 Python Language:

Python is widely recognized as one of the most versatile and powerful programming languages, and its advantages in voice recognition are manifold. One of the key strengths of Python in this domain is its extensive range of libraries and frameworks specifically tailored for speech processing and recognition tasks. Libraries such as SpeechRecognition and pocketsphinx provide developers with easy-to-use tools for converting speech to text, enabling seamless integration of voice input into applications. Moreover, Python's simplicity and readability make it ideal for rapid prototyping and experimentation, allowing developers to quickly iterate and refine voice recognition algorithms. Additionally, Python's strong community support means that there are numerous tutorials, guides, and forums available for assistance, facilitating the development process. Overall, Python's versatility, extensive libraries, and supportive community make it a preferred choice for voice recognition applications, empowering developers to create innovative and user-friendly experiences.

3.2.4 MySQL:

MySQL is a robust and widely-used relational database management system (RDBMS) that offers powerful capabilities for storing, managing, and retrieving data. Its popularity stems from its reliability, scalability, and ease of use, making it an ideal choice for a wide range of applications. For command history and to-do list management, MySQL provides a structured and efficient way to store and organize data. By creating tables to represent command history and to-do list items, users can easily add, retrieve, update, and delete records using SQL queries. MySQL's support for transactions ensures data integrity and consistency, while its indexing capabilities optimize query performance, especially when dealing with large datasets. Whether it's tracking command histories for analysis or managing tasks efficiently in a to-do list, MySQL empowers users with a reliable and efficient solution for data storage and retrieval.

3.3 User Flow

- **Initialization:** The user launches the voice Desktop Assistant application, initial database connection and setup of tables is done.
- **Interaction Initiation:** The user chooses to interact with the assistant either through text input or voice command mode.
- **Text Input Mode:** If the user selects text input mode, they type their query into the chat interface and submit it.
- **Voice Command Mode:** If the user selects voice command mode, they speak their query into the microphone, which is captured by the application.
- **Voice Recognition:** The application employs voice models and python libraries to transcribe the user's spoken words into text.
- **Running Various Commands:** The user can run various commands like opening applications and websites, sending messages via WhatsApp and email, managing to-do lists, retrieving news updates, playing music, doing web searches, telling jokes or sharing fun facts, providing system information (e.g., current time) and more.

3.4 Database Structure

SQL database is used to store query history and the To-do list as they are two features which require the system to constantly update, save, edit and remove data seamlessly according to the users needs. Both are stored in tables as shown in **Fig. 1** below. Data can easily be updated to and retrieved from the database by the code with the help of connector libraries like mysql-connector etc. The initial tables are set up and initialized when the program is run and isn't altered if it already exists, ensuring that data isn't lost between different occurrences of usage.

```
mysql> use queries;
Database changed
mysql> select * from todo;
+-----+-----+
| task_number | task_text                |
+-----+-----+
|          1 | what's the cricket match |
|          2 | wash my clothes          |
|          3 | wash my clothes          |
|          4 | clean the dishes         |
+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

Fig. 1: Example of database structure for To-do list

3.5 Results

The final user interface of the project is shown below (**Fig. 2**). A toggle is present to activate voice command functionality after which the user can start interacting with the application using their voice. The main graphical interface is prepared using the streamlit library.

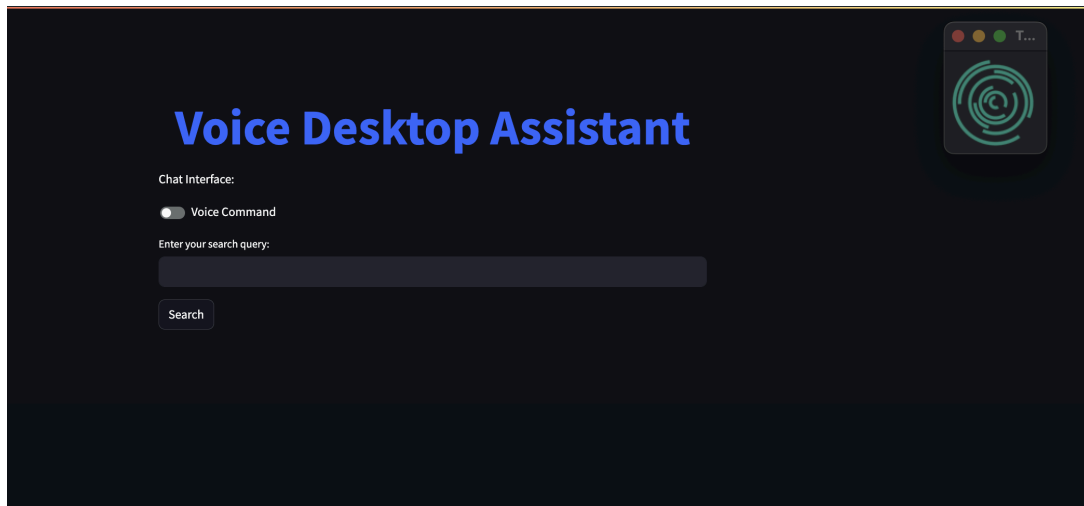


Fig. 2: User interface of the application

The application interacts with the user and provides input based on the query of the user. An example of how the application responds to the command is shown in **Fig. 3**. In the figure, the user asks the assistant to tell them the news and the assistant responds appropriately with the top 5 news headlines of the day.

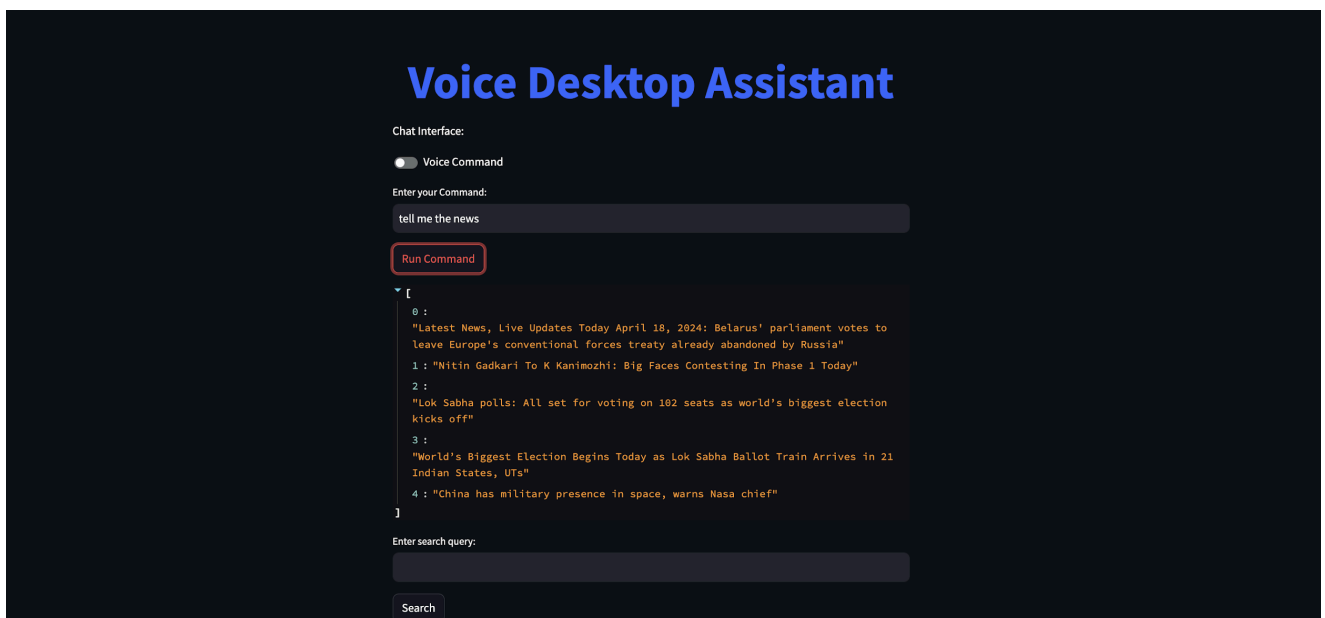


Fig 3: Example of functionality: News Update

The application also integrates additional Tkinter windows to provide additional functionality. Two examples of this are the Mini Assistant and E-mail assistant. The Mini Assistant also uses the pillow library to display a GIF to remind the user when they are on a different tab or screen that the assistant is running in the background. The E-mail assistant is called when the user gives the command to send an email.

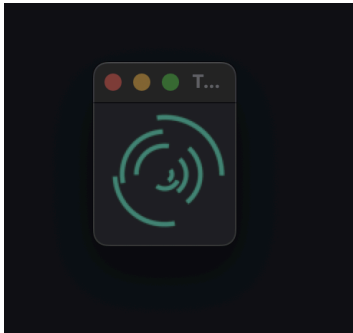


Fig 4: Mini Assistant

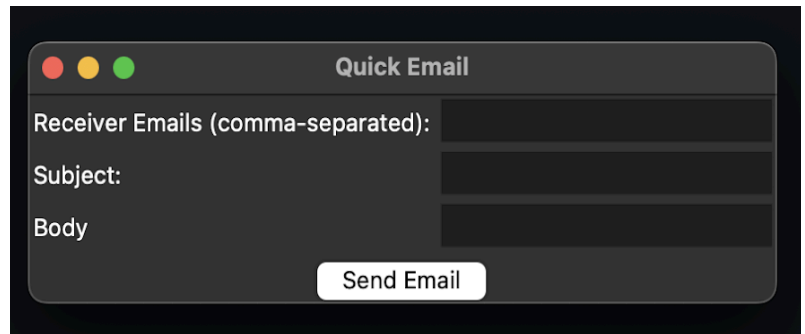


Fig 5: Email Assistant

CHAPTER 4

4.1 Future Scope

Looking to the future, additional measures that can be taken to improve the functionality of the system include:

- Database Enrichment: Expand database for broader and updated information coverage.
- Improved User Interface (UI): Enhance UI for better aesthetics and usability.
- Enhanced Usability: Implement dynamic features for personalized user experience.
- Integration of Admin Dashboard: Introduce Admin Dashboard for performance insights and analytics.
- Multi-Language Support: Extend language support for broader user accessibility.
- Integration with Smart Devices: Explore integration with smart home devices for expanded functionalities.
- Continuous Learning and Improvement: Incorporate feedback-driven learning and ML for enhanced performance.

4.2 Conclusion

In conclusion, the Voice Desktop Assistant project has laid a strong foundation for creating an intelligent and interactive companion that enhances user productivity, accessibility, and convenience in desktop computing environments. With its current set of features and functionalities, coupled with the potential for future expansion and improvement outlined above, JARVIS is poised to evolve into a versatile and indispensable tool for users seeking efficient and intuitive assistance in their daily tasks. By embracing advancements in technology and incorporating user-centric design principles, the project aims to deliver a seamless and enriching user experience while continuing to innovate and adapt to the evolving needs of its users.

REFERENCES

- **Vosk(Offline Voice Recognition):** www.alphacephei.com/vosk/install
- **MySQL Documentation:** www.dev.mysql.com/doc/
- **Streamlit hosting documentation:** www.docs.streamlit.io
- **For voice commands :** Www.people.csail.mit.edu/hubert/pyaudio/docs/
- **Python commands:** www.docs.python.org/3/library/os.html