

Chapter-1

INTRODUCTION

1.1 Introduction:

Almost all tasks are now digitalized in today's world. Voice searches have surpassed text searches. Web searches conducted via mobile devices have only recently surpassed those conducted via computer, and analysts predict that 50% of searches will be conducted via voice by 2024. Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to handle your email. Detect intent, extract critical information, automate processes, and provide personalized responses. In recent years, several researchers have become interested in the recognition of human activities. The desktop's virtual assistant in Python is a software programme that assists you with day-to-day tasks such as showing the weather report, creating reminders, making shopping lists, and so on. They can respond to commands via text (as in online chat bots) or by voice. This system is intended for use on desktop computers. Virtual assistant software boosts user productivity by managing routine tasks and providing information from online sources. In this project, we propose a voice recognition system that recognizes human activities by utilising an NLP algorithm. Voice is a form of communication in which users can communicate with one another. Automatic Speech Recognition (ASR), also known as voice recognition, recognizes spoken words and phrases and converts them to computer readable formats. It accepts user input in the form of voice or text, processes it, and provides feedback to the user in a variety of ways, such as the action to be taken or the search result. As a result, distinguishing spoken words from background noise in audio is an additional challenge.

1.2 Motivation:

The main purpose of this project is to build a program that will be able to service to humans like a personal assistant. This is an interesting concept and many people around the globe are working it. Today, time and security are the two main things to which people are more sensitive, no one has the time to spoil; nobody would like their security breach, and this project is mainly for those kinds of people.

This system is designed to be used efficiently on desktops. Virtual Assistants software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user. This project was started on the premise that there is enough openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

Chapter-2

PROBLEM STATEMENT

2.1 Literature Survey:

Kim, S. et al. (2023). Efficient Distillation for Information Retrieval Tasks with Large Transformer Models. Briefly summarize the paper's focus on efficient deployment of large neural models for information retrieval while maintaining performance. Mention the successful distillation approach and its effectiveness.[1]

The investigation into user experience and motivation in customer service chatbots. Highlight the likely examination of factors like usability, satisfaction, and perceived usefulness. [2].

Briefly summarize the proposal of Embed Distil for geometric knowledge distillation in information retrieval. Mention its focus on improving efficiency and effectiveness by distilling knowledge into smaller models. [3]

Bonifacio, L. et al. (2022). Briefly summarize the highlighted transformation in the Information Retrieval field due to large pretrained transformer models. Mention the role of the MS MARCO dataset in enabling zero-shot transfer learning. [4]

The introduction of Gradio, a platform for sharing and testing machine learning models. Highlight its features and capabilities for user-friendly deployment and interaction. [5]

Briefly summarize the proposed open-source approach for building an LLM News Bot Investigator. Mention its unique aspects like open-source framework, interactive interface, and comparative model study. [6]

This project proposed a novel open-source approach to desktop Assistant using custom knowledge from the content. It is unique in several ways:

1. Briefly summarize the key points and findings of each paper.
2. Use concise and clear language.
3. Maintain a consistent format throughout the literature survey.
4. Ensure proper citation formatting for each reference.

2.2 Need of Work:

In today's digital age, the demand for streamlined productivity and personalized user experiences has never been higher. The complexity of modern work environments, coupled with the abundance of tasks and information, necessitates innovative solutions to simplify workflows and enhance efficiency. Enter the Desktop Assistant project: a proactive and adaptable software solution designed to meet the diverse needs of users across various computing environments. Leveraging state-of-the-art technologies such as artificial intelligence, natural language processing, and machine learning, the Desktop Assistant serves as a central hub for accessing information, managing tasks, and executing commands seamlessly. By integrating with the user's desktop environment, the Assistant becomes an indispensable tool for automating routine tasks, providing timely reminders, and delivering personalized recommendations tailored to individual preferences and behaviours. Furthermore, the project addresses the growing demand for cross-platform experiences, as the Assistant bridges the gap between disparate devices and services, offering a unified interface for interacting with applications, data sources, and services. Through its intuitive interface and intelligent capabilities, the Desktop Assistant empowers users to navigate their computing environments with ease, enabling them to focus on high-value activities and maximize productivity. In essence, the Desktop Assistant project not only meets the immediate need for productivity-enhancing tools but also anticipates future trends towards intelligent automation and personalized assistance in the digital realm.

2.3 Problem Statement

In response to the challenges of desktop computing, a sophisticated desktop assistant is needed. This intelligent solution addresses issues of task overload, information management, multitasking inefficiencies, and workflow automation. Prioritizing user customization, communication integration, and robust security, the assistant aims to redefine and enhance the overall desktop computing experience for individuals and professionals alike.

2.4 Objectives:

1. **Information Accessibility:** Virtual assistants act as quick sources of information on websites and mobile platforms, enhancing user experience through accessibility.

2. **Time Efficiency in Research:** They streamline research tasks, enabling users to delegate topic-specific research and focus on other activities, thereby enhancing productivity.
3. **Personal Organization:** Serving as personal organizers, virtual assistants remind users of important dates such as tests, birthdays, or anniversaries, aiding in organization and preparedness.
4. **Voice Search Efficiency:** Virtual assistants facilitate faster information retrieval through voice searches, thanks to efficient voice recognition technology, contributing to overall efficiency and user satisfaction.

Chapter-3

DESIGN DETAILS

3.1 System Architecture:

The Desktop Assistant project is structured around four main modules, each serving a specific purpose in facilitating user interaction and providing intelligent assistance within the desktop environment.

Module 1: Pre-processing

This module is responsible for handling incoming data and user inputs, ensuring they are cleaned, normalized, and structured appropriately for further processing. It involves parsing user inputs from various sources such as keyboard, mouse, or voice commands, cleaning and normalizing the data to remove noise and inconsistencies, and performing language processing tasks such as tokenization, entity recognition, and contextual understanding.

Module 2: Domain Terminology Extraction

The Domain Terminology Extraction module focuses on extracting domain-specific terms and concepts relevant to the user's context or workflow. It leverages techniques such as natural language processing, named entity recognition, and domain-specific dictionaries to identify and categorize key terms related to the user's domain of interest. This module plays a crucial role in understanding user queries and providing tailored assistance based on the specific domain or industry.

Module 3: Supervised Learning with Back Propagation Algorithm

The Supervised Learning module utilizes machine learning algorithms, particularly the Back Propagation Algorithm, to train models for various tasks such as intent recognition, sentiment analysis, or recommendation. It involves collecting labelled training data, extracting relevant features, and training neural network models using the backpropagation algorithm to minimize prediction errors. This module enables the Desktop Assistant to learn from past interactions and adapt its behaviour to better serve the user's needs over time.

Module 4: Filtering of Results

The Filtering of Results module is responsible for filtering and ranking the results obtained from the previous modules based on relevance, importance, or user preferences. It may incorporate techniques such as rule-based filtering, semantic similarity analysis, or collaborative filtering to prioritize and present the most relevant information or

recommendations to the user. This module ensures that the Desktop Assistant delivers accurate and meaningful results, enhancing the overall user experience.

3.3 System Design Diagrams

3.3.1 Data Flow Diagram

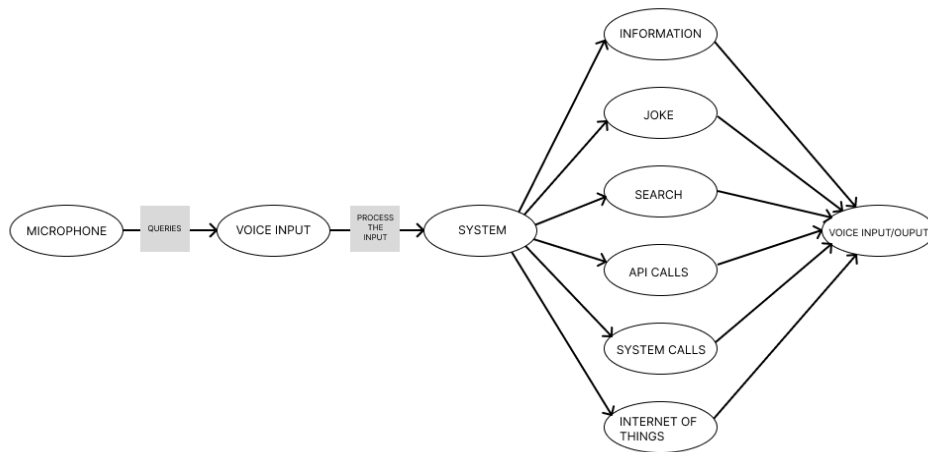


Fig 3.3.1 Flow Diagram

User speaks queries or interacts with the system through a microphone. The "Microphone Queries Voice Input Process System" component converts the audio input into text. The text is then further processed and analysed to understand the user's intent. Use "Search" to access information from online sources like the internet. Utilize "API Calls" to interact with other systems and retrieve relevant data. Leverage the "Internet of Things" to gather information from connected devices. The gathered information is processed and analysed. The "Joke" component might be used to generate humorous responses in specific contexts. Displayed on a screen if its text based. Converted into audio and spoken back to the user through a speaker. Used to execute specific actions based on the user's intent (e.g., controlling smart home devices). This is a high-level explanation based on limited information. More details about the specific components and their interactions would be needed for a more precise understanding. The architecture diagram suggests a modular approach, where different components handle specific tasks, potentially improving flexibility and maintainability. The inclusion of "Joke" suggests the system might have the ability to provide humorous responses, adding an element of personality.

3.3.2 Use Case Diagram:

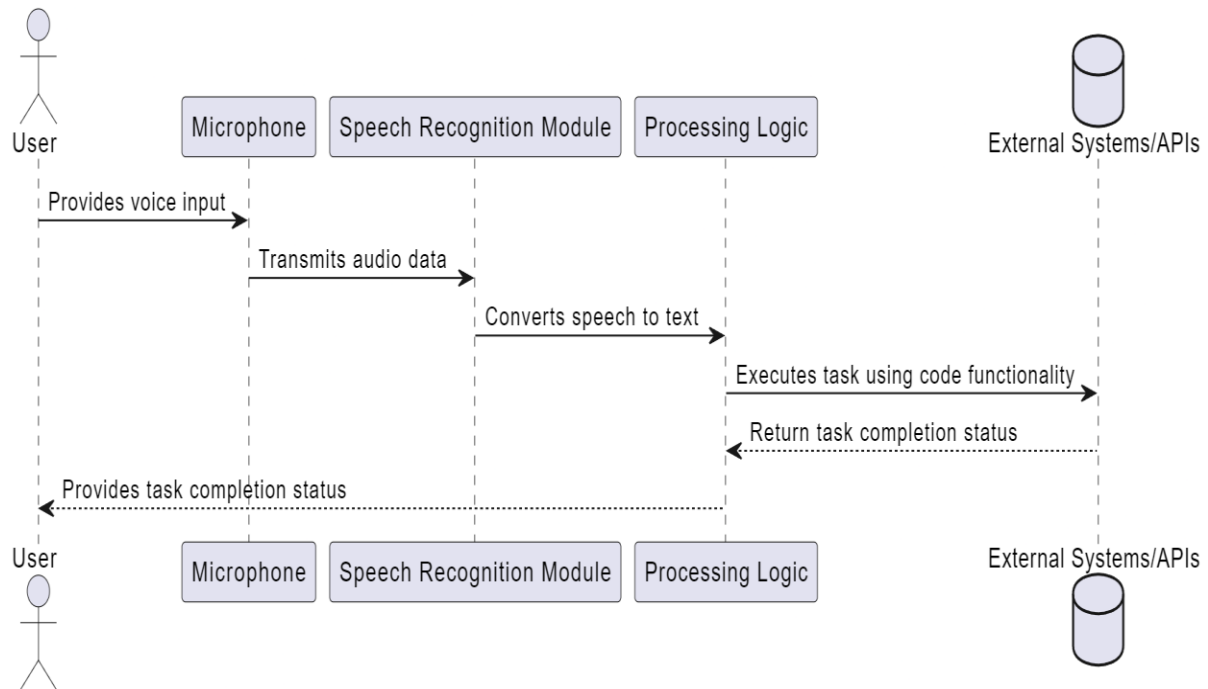


Fig 3.3.2 Use Case Diagram

In UML, use-case diagrams model the behaviour of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the use.

3.3.3 Class Diagram:

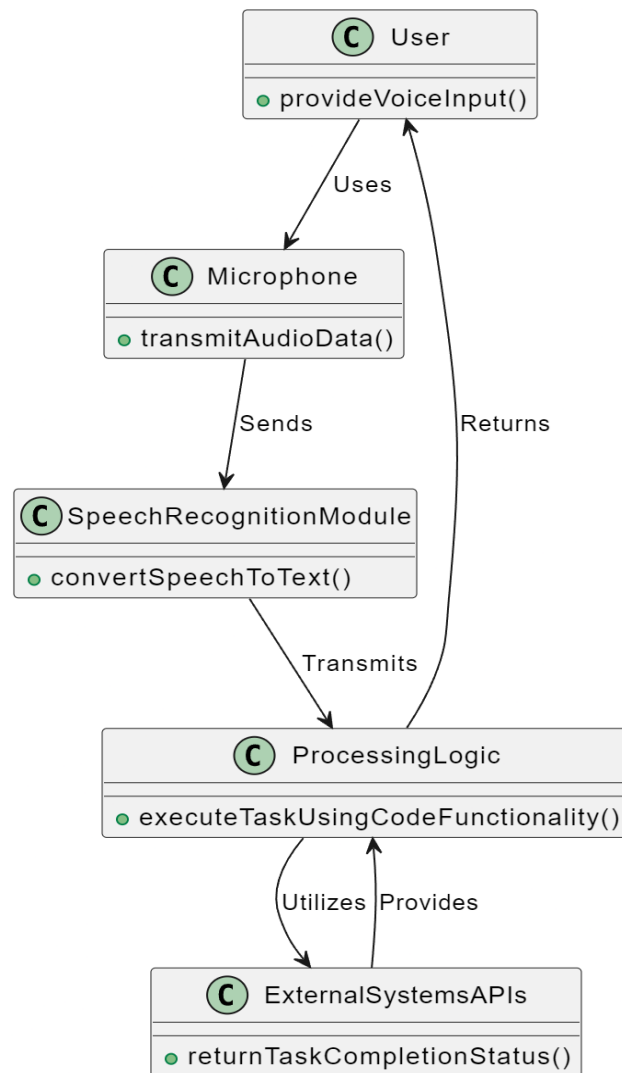


Fig 3.3.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format.

Chapter – 4

IMPLEMENTATION

4.1 System Requirement

4.4.1 Hardware Requirement

- **Processor:**

A modern CPU with at least 2 cores and a clock speed of 2.0 GHz is recommended. For more demanding assistants or if you plan to use the assistant for resource-intensive tasks like video editing, a quad-core processor or higher is recommended.

- **RAM**

RAM: 4GB of RAM is the minimum requirement, but 8GB or more is recommended for better performance.

- **Storage:**

Storage: At least 16 GB of free storage space is needed for the assistant's software and data. A solid-state drive (SSD) is recommended for faster performance.

- **Microphone:**

A good quality microphone is essential for accurate voice recognition. A USB microphone is a good option for flexibility.

- **USB microphone**

Speakers or headphones: You'll need speakers or headphones to hear the assistant's responses. A headset is recommended if you plan to use the assistant for making calls or video conferencing.

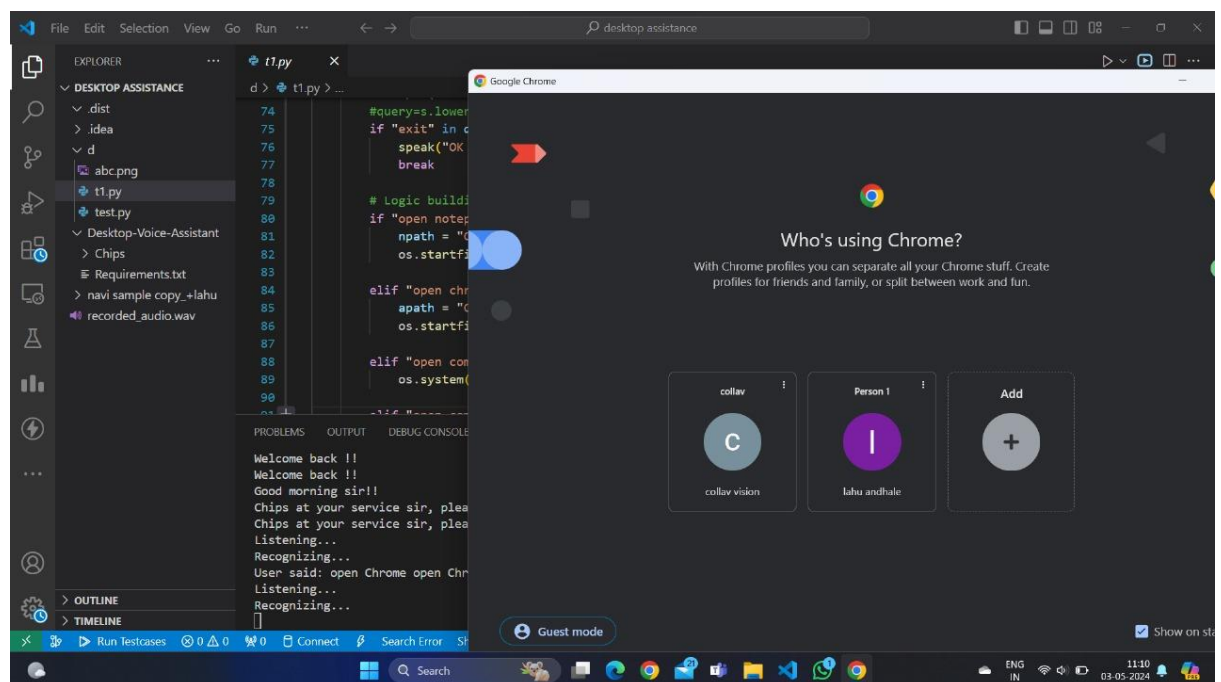
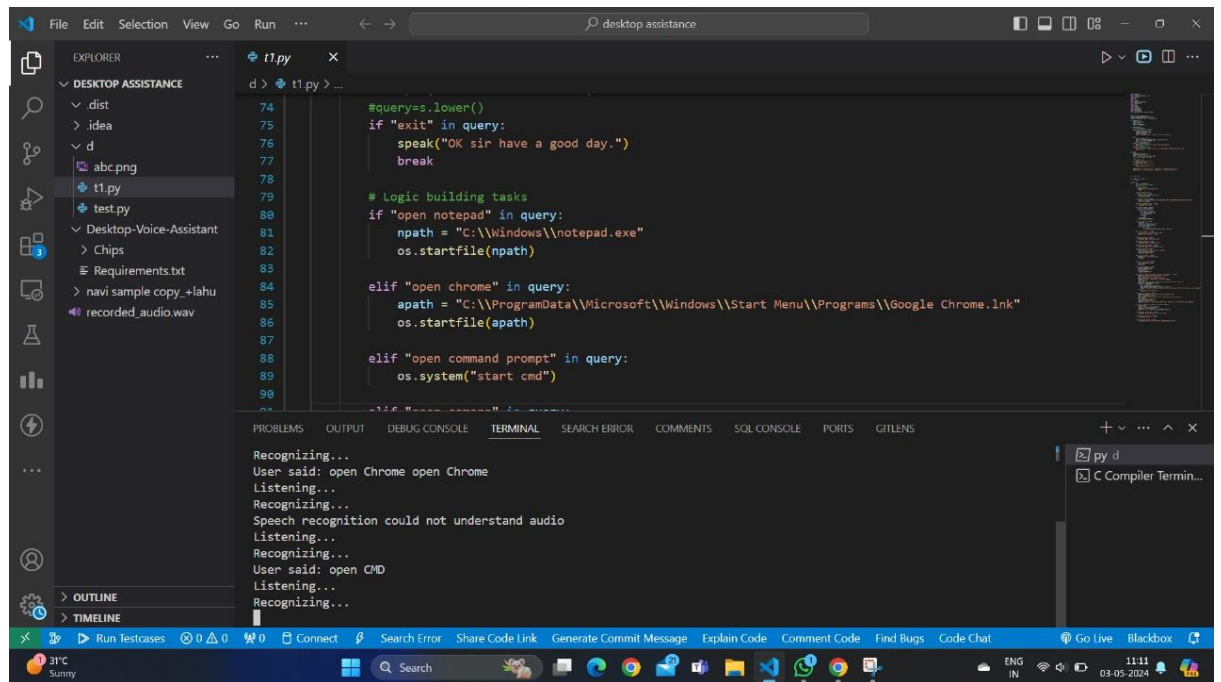
4.4.2 Software Requirement

- **Operating system:** The assistant should be compatible with your operating system (e.g. Windows). Internet connection: An internet connection is required.
 - **Development Environment:** Visual Studio Code, PyCharm.
 - **Programming Language:** Python 3.9.6
- **Required Python Packages:**
- **Pyttsx3:** Enables text-to-speech functionality, allowing the Desktop Assistant to communicate with users audibly.
 - **Speech Recognition:** Provides speech recognition capabilities, enabling the assistant to understand spoken commands and queries.
 - **Wikipedia-API:** Allows the assistant to access and retrieve information from Wikipedia articles.
 - **Paudie:** Essential for audio input and output, facilitating speech recognition and text-to-speech conversion.
 - **Web browser:** Enables the assistant to launch web pages or search results in the default web browser, enhancing its capabilities for retrieving online information.

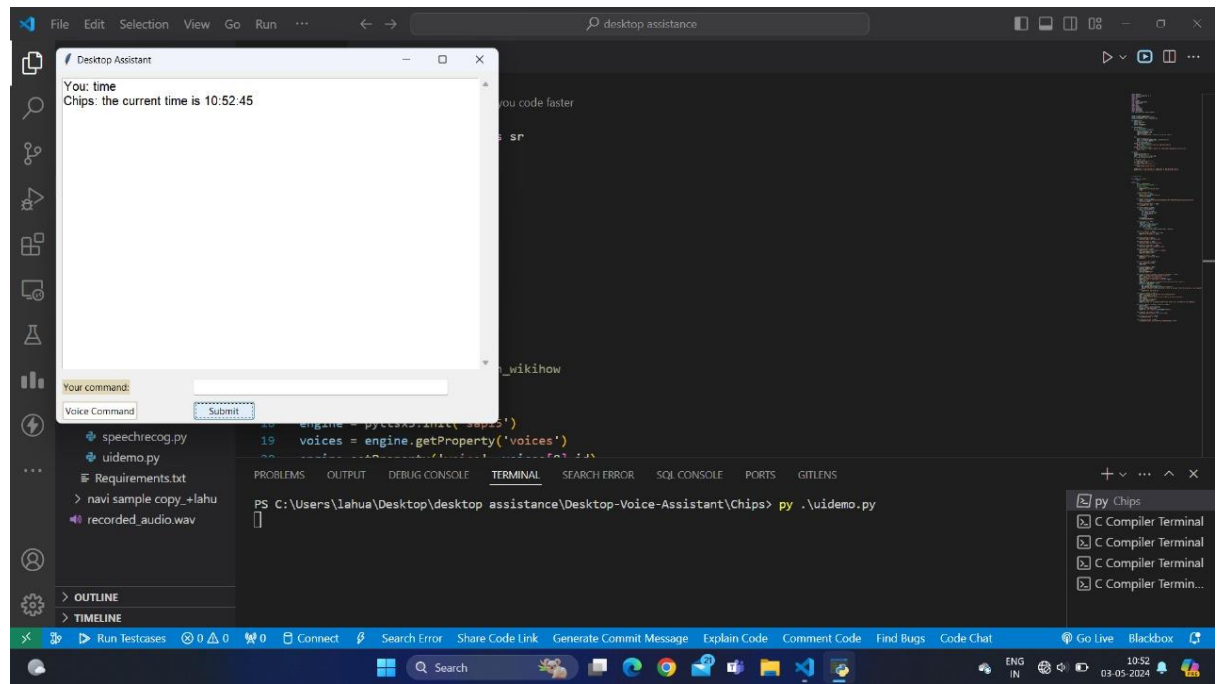
Chapter -5

EXPERIMENTAL RESULTS

Desktop Assistant



Desktop Assistant



Chapter – 6
CONCLUSION

6.1 Conclusion:

Virtual Assistants for the desktop that use Python are a very effective way to organise your schedule. Today, numerous Smart Personal Digital Assistant applications are available for a variety of device platforms. Because they have access to all of your Smartphone's resources, these new Software Applications outperform PDA devices. Because they are more portable and can be used at any time, virtual assistants are more dependable than human personal assistants. Because they have access to the internet, they have access to more information than any other assistant. The Python-based virtual assistant on the desktop is dependable and provides information in a user-friendly manner.

6.2 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 plans include integrating our virtual assistant with mobile using React Native to provide a synchronised experience between the two connected devices. Further, in the long run, our virtual assistant is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT.

Chapter- 7
REFERENCES

7.1 References:

1. Sangpal, R., Gawand, T., Vaykar, S., & Madhavi, N. (2019, July). JARVIS: An interpretation of AIML with integration of gates and Python. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (Vol. 1, pp. 486-489). IEEE.
2. Othman, E. S. (2017). Voice Controlled Personal Assistant Using Raspberry Pi. International Journal of Scientific & Engineering Research, 8(11), 1611-1615.
3. Mittal, Y., Toshniwal, P., Sharma, S., Singhal, D., Gupta, R., & Mittal, V. K. (2015, December). A voice controlled multifunctional smart home automation system. In 2015 Annual IEEE India Conference (INDICON) (pp. 1-6). IEEE.
4. Pandey, A., Vashist, V., Tiwari, P., Sikka, Makkar, P. Smart Voice Based Virtual Personal Assistants with Artificial Intelligence.
5. K. Gunasekaran, V. Vinoth Kumar, A. C. Kaladevi, T. R. Mahesh, C. Rohith Bhat and K. Venkatesan, "Smart Decision-making and Communication Strategy in Industrial Internet of Things," in IEEE Access, doi:10.1109/ACCESS.2023.3258407
6. Venkatesan, V.K.; Ramakrishna, M.T.; Izonin, I.; Tkachenko, R.; Hawryluk, M. Preprocessing with Ensemble Machine Learning Technique for the Early Detection of chronic kidney disease. Appl. Sci. 2023, 13, 2885.