ASTRA: The Virtual Assistant for Personal Computer

Ms M Vineela[1], Madishetty Nandini[2], Ranga Akshitha[3]
[1]Associate professor, Department of CSE, Bhoj Reddy Engineering College for Women, India
[2,3]B.Tech Students, Department of CSE, Bhoj Reddy Engineering College for Women, India

ABSTRACT
In today's fast-paced digital world, managing routine tasks efficiently on Personal Computers has become essential for enhancing productivity and convenience. Virtual assistants like Alexa and Cortana provide some functionalities but lack the customization and tailored interface optimized for Personal Computer (PC) environments. ASTRA, a voice-activated virtual assistant, is designed to address these gaps by offering a robust solution that combines real-time voice recognition, an intuitive graphical user interface (GUI), and versatile functionalities.
Developed using Python, Flask, and advanced speech recognition technologies, ASTRA empowers users to perform tasks such as web searches, weather updates, YouTube access, app launching, and image capturing seamlessly. With its innovative features, ASTRA not only simplifies day-to-day activities but also ensures accessibility for all users. By integrating real-time data services and automation, this model redefines the Personal Computer experience, providing a comprehensive and user-friendly platform to optimize workflows and enhance overall efficiency.
Keywords*: Virtual Assistant, Speech Recognition, Natural Language Processing (NLP), Task Automation, Personal Computer (PC) Assistant.*

1. INTRODUCTION

The project aims to develop a private assistant for personal computers, named ASTRA. ASTRA draws its inspiration from virtual assistants like Google Assistant for Android and Siri for iOS. It provides a user-friendly interface to complete a variety of tasks using well-defined commands. Users interact with ASTRA primarily through voice commands, enabling a hands-free experience for managing daily activities.

As a private assistant, ASTRA supports tasks such as engaging in general human-like conversations, conducting searches, fetching the latest news, providing live weather updates, setting reminders, and taking screenshots or photos. ASTRA uses a device's microphone to receive voice requests, while the output is delivered through the system's speaker. It combines multiple technologies, including voice recognition, voice analysis, and natural language processing, to execute user queries efficiently.

The workflow begins with ASTRA converting natural language audio signals into digital data, which is then processed using various models to transcribe audio into text. Specific keywords trigger corresponding functionalities. For example, the "weather" keyword fetches live weather information, and the "play" keyword plays YouTube videos.

ASTRA's tasks include retrieving information from Wikipedia, reading top news from sources like Times of India, fetching live weather updates for a specified city, performing computational tasks, capturing photos via the webcam, and more. It utilizes APIs such as OpenWeatherMap for weather updates and libraries like speech_recognition for speech-to-text conversion,

pyttsx3 for text-to-speech conversion, and pywhatkit for YouTube or web search operations. Python's robust libraries and modular design enable seamless integration of these features, making ASTRA a versatile and efficient virtual assistant for personal computers.

2. LITERATURE SURVEY

T. D'Orazio et al. (2021) conducted a comparative analysis of the most widely used smart voice assistants, namely Amazon Alexa, Google Assistant, and Microsoft Cortana. Their study assessed these assistants based on user interactions and perception. The findings indicated that Alexa and Google Assistant performed better than Cortana, with minimal differences between the two in terms of accuracy and response quality. The research emphasized the increasing role of artificial intelligence in human-machine interaction, with applications extending to education, healthcare, robotics, home automation, and security. The study provided insights into the strengths and weaknesses of commercial voice assistants, setting a foundation for the development of customized AI-driven assistants like ASTRA.

Rijo Benny et al. (2024) introduced an OpenAI-enhanced personal desktop assistant, designed to improve human-computer interaction with a focus on security and personalized user experience. The study compared user satisfaction rates across major voice assistants, reporting that Amazon Alexa and Google Assistant had the highest adoption and satisfaction levels, while Microsoft Cortana lagged behind. Their work emphasized the privacy concerns associated with cloud-based virtual assistants, motivating the development of a local desktop assistant that operates without reliance on external cloud services. By leveraging Python programming and OpenAI models, their system achieved robust natural language processing (NLP) capabilities, making it more adaptable to user preferences. This research aligns with ASTRA's objective of creating a personalized PC-based assistant with enhanced privacy and AI-driven functionalities.

Jash Vora et al. (2021) developed JARVIS: A PC Voice Assistant, inspired by popular virtual assistants such as Google Assistant and Siri. The study focused on providing a user-friendly desktop interface capable of executing various tasks through voice commands, including web searches, news retrieval, weather updates, email automation, and language translation. The assistant utilized voice recognition, analysis, and natural language processing to interpret user inputs and generate appropriate responses. The system was implemented using Python, leveraging microphone input and speaker output for a seamless voice-driven interaction experience. The research demonstrated the feasibility of PC-based virtual assistants, reinforcing the concept behind ASTRA's development.

3. METHODOLOGY

ASTRA is built using a **modular architecture** that ensures flexibility, scalability, and seamless task execution. The methodology involves the following components:

3.1 System Architecture

ASTRA follows a three-tier architecture:

- **Frontend:** Developed using Tkinter for GUI, ensuring an interactive and user-friendly interface.
- **Backend:** Implemented with Python and Flask for handling commands and executing user requests.
- **Database:** MySQL is used to store user interactions and chat history, improving personalization and performance.

- **APIs & Libraries:** Integration with various APIs and libraries such as SpeechRecognition for voice input processing, Pyttsx3 for text-to-speech conversion, Pywhatkit for automating searches, OpenCV for image capturing, and Webbrowser for launching web applications.

Existing system

Current virtual assistant technologies, like **Alexa** and **Cortana**, serve as the foundation for modern AI-driven personal assistants but are largely designed for specific platforms and environments.

Alexa was initially created for smart home devices, offering functionalities like controlling appliances, playing music, and providing weather updates. While Alexa has evolved to be integrated into mobile platforms, it still lacks the ability to provide deep integration with personal computers. Alexa excels in controlling Internet of Things (IoT) devices but falls short in offering a comprehensive solution for tasks like file management, application launching, or providing a customized GUI that would cater to personal computer workflows.

Cortana, integrated into the Windows operating system, provides users with a basic assistant capable of handling tasks like web searches, reminders, and opening apps. However, **Cortana's integration with personal computers** is limited in that it lacks the specific user interface optimizations needed for more advanced desktop tasks. Cortana's **GUI** is not as customizable or interactive as required for personal computer users, limiting its usability for tasks that go beyond simple operations like setting reminders or searching the web.

Both **Alexa** and **Cortana** have significant constraints when it comes to being used effectively in personal computer environments, especially for users who require both simple and advanced functionalities integrated into one cohesive system.

Proposed System

The ASTRA Virtual Assistant is specifically designed to overcome the limitations of existing systems by providing a comprehensive solution tailored for personal computer use. Unlike assistants like Alexa and Cortana, which are optimized for smart homes and mobile platforms, ASTRA integrates advanced features that cater specifically to desktop environments. Built using Python, Tkinter, and other libraries, ASTRA combines speech recognition with a customized Graphical User Interface (GUI), offering a more versatile and user-friendly experience for personal computer users.

Fig 1 An example interface

Key features of ASTRA include voice recognition, which allows users to execute commands such as opening applications, conducting web searches, and checking weather reports. Additionally, ASTRA includes text-to-speech functionality, converting text-based responses into speech and enhancing accessibility for users with disabilities. Another unique feature of ASTRA is its image capture capability, using the webcam to take photos based on user commands. This feature adds a visual element to the assistant's functionality. Task automation is also a core strength of ASTRA, allowing it to handle tasks such as setting reminders, managing applications, and providing real-time weather updates, thus streamlining the user's workflow. The assistant is further enhanced by web and YouTube search capabilities, which are integrated with APIs to search the internet and play videos directly from voice or text input. ASTRA also supports user personalization, remembering details such as the user's name and preferences to offer more customized responses, making interactions more natural and engaging.

4. RESULTS

The image showcases the web interface of "Astra," a chatbot or application. At the center, a card displays the text "Welcome to Astra". Below this, two blue buttons labeled "Start" and "Stop". On the right side of the page, there is a floating "Help" button it shows a guide tour.

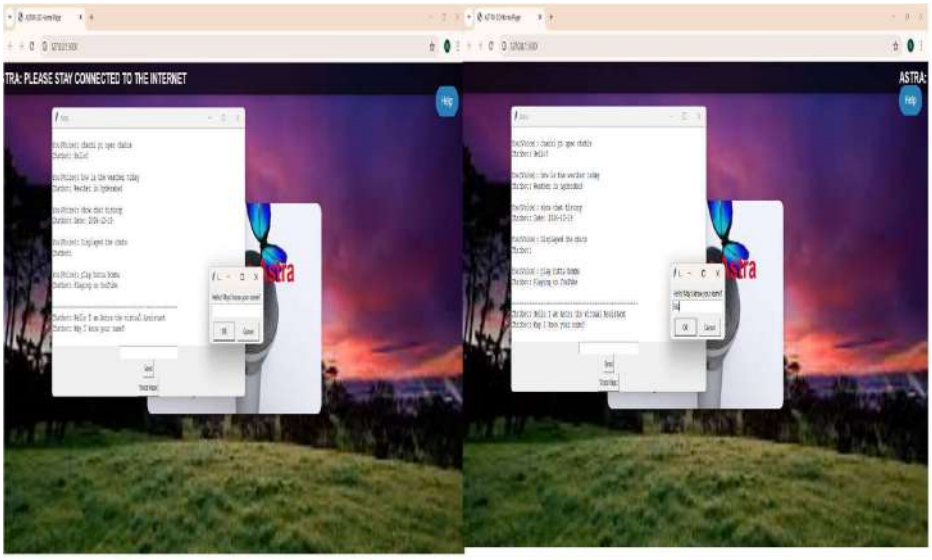

Fig 2 First Visit Appearance

After clicking the start button it executes the astra.py code automatically and shows astra GUI (graphical User Interface) successfully. If it is first time, asks for the user name. It remembers the name of the user and address with name for the next visit.

: It starts with dialog: “Hello how can I assist you today”

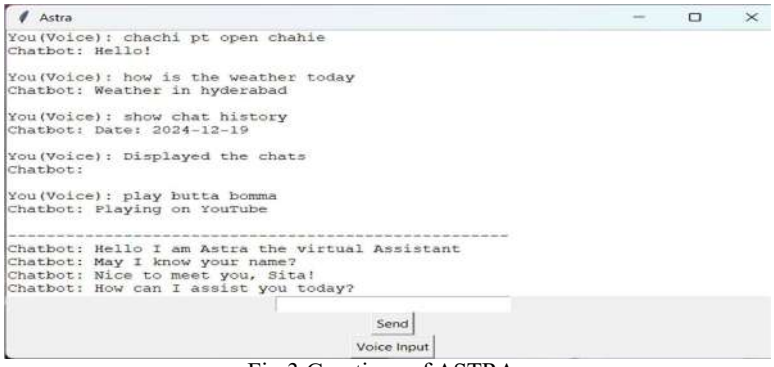

Fig 3 Greetings of ASTRA

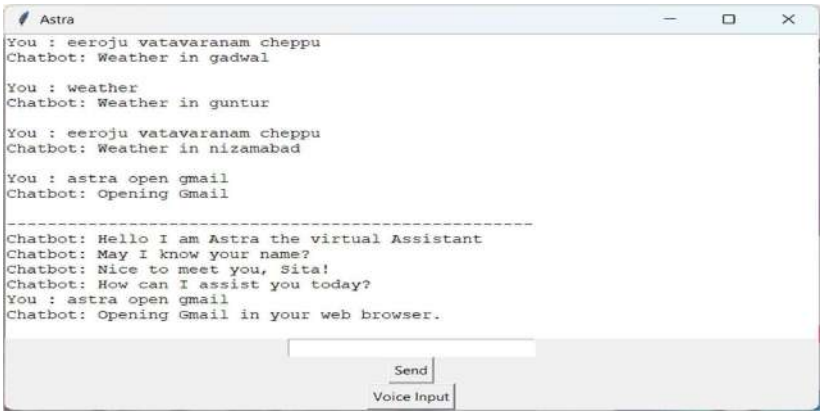

Fig 4 Open Gmail Input

Output: Opening Gmail in user browser Successfully.

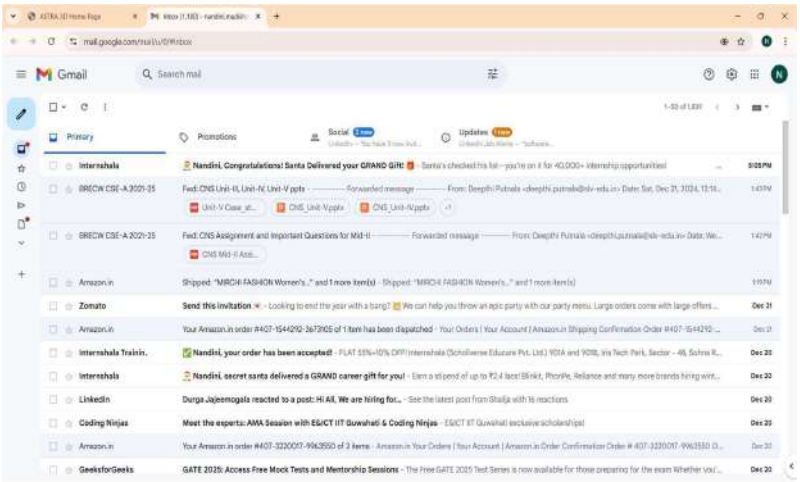

Fig 5 Open Gmail Output

Input:  or  Astra open ChatGPT

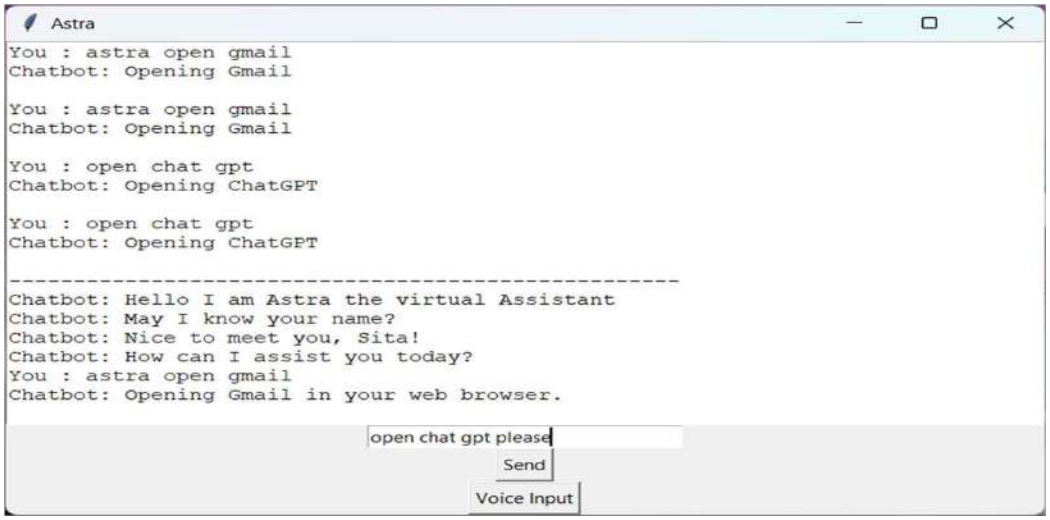

Fig 6 Open ChatGPT Input

Output : Opening ChatGPT

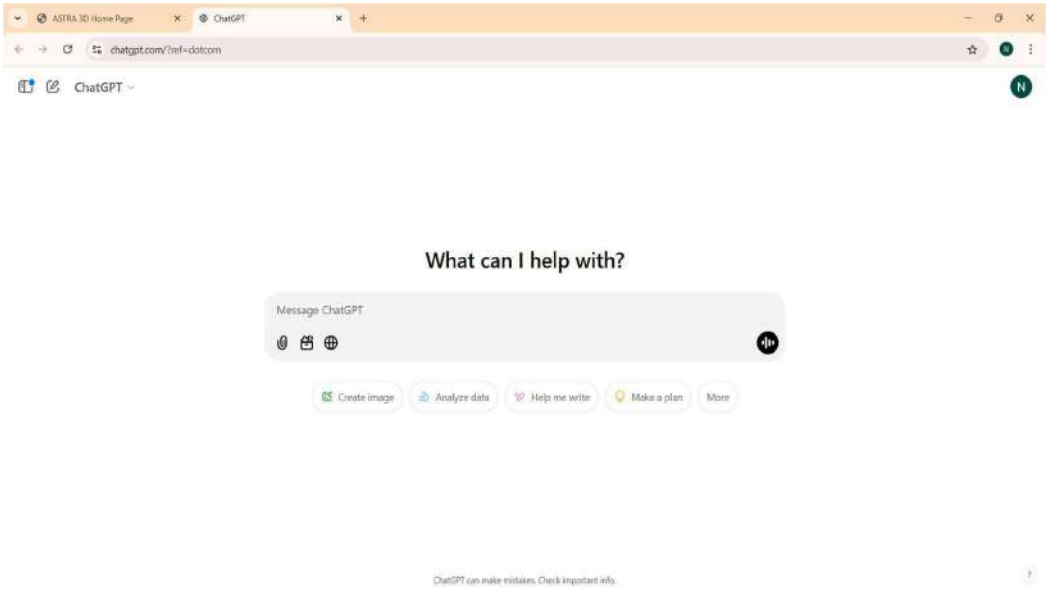

Fig 7 Open ChatGPT Output

Input:  or  Current prime minister of india.

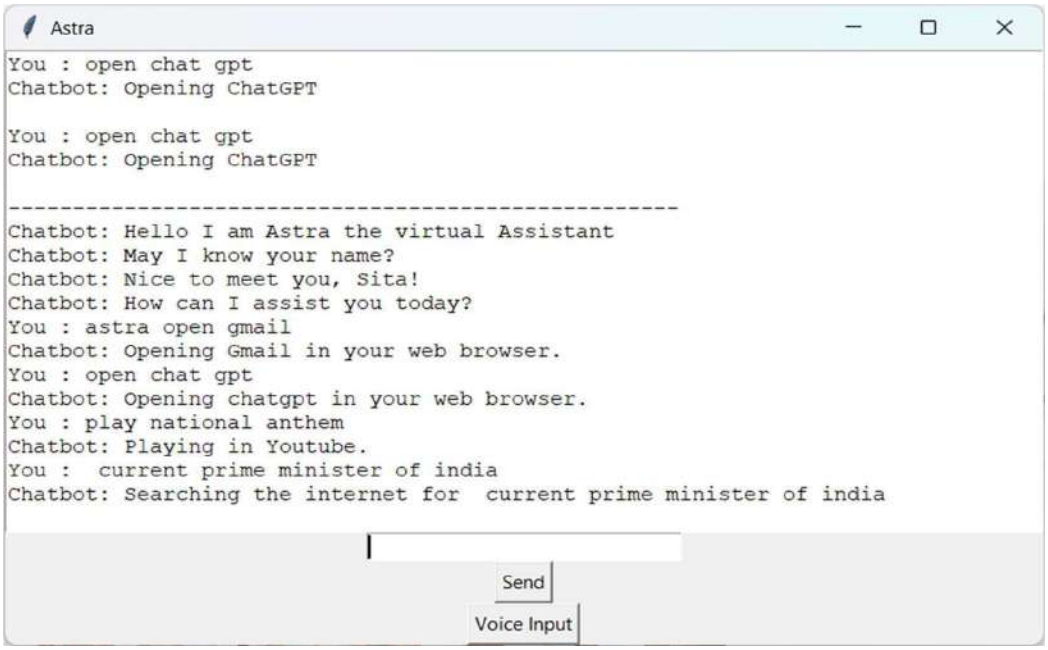

Fig 8 Internet Search Input

Output: Display the result in the google browser.

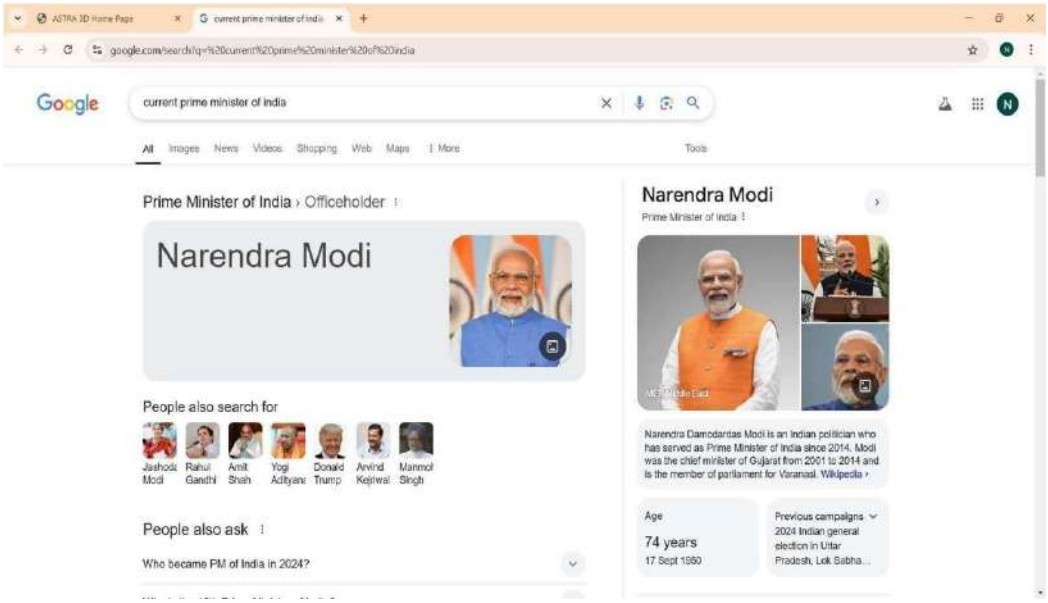

Fig 9 Internet Search Output

Input :  or  Take a screenshot

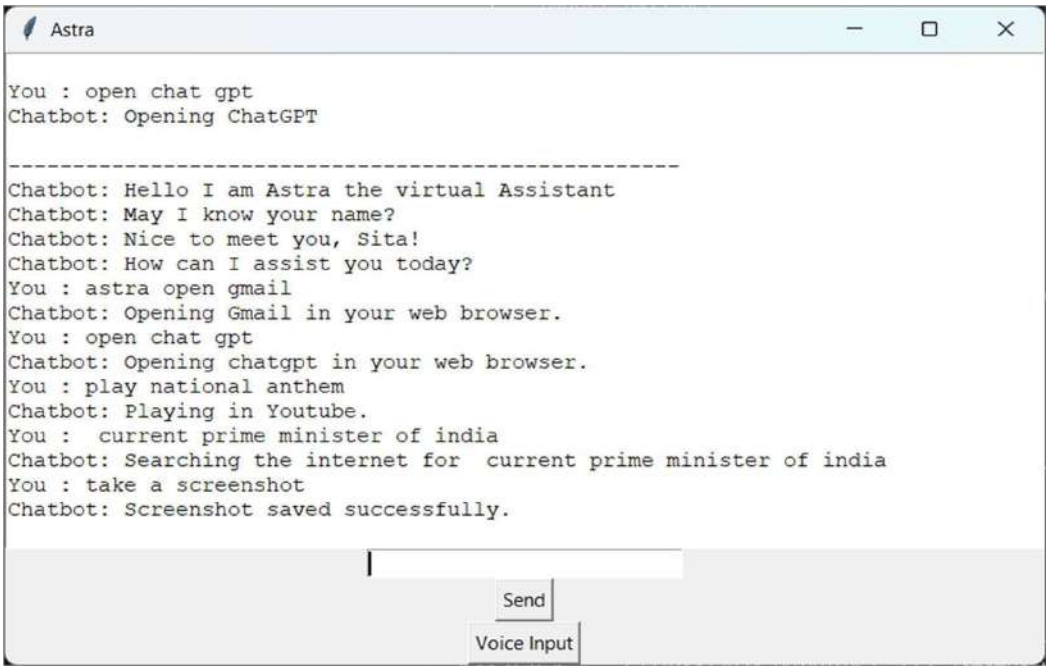

Fig 10 Take Screenshot Input

Output:

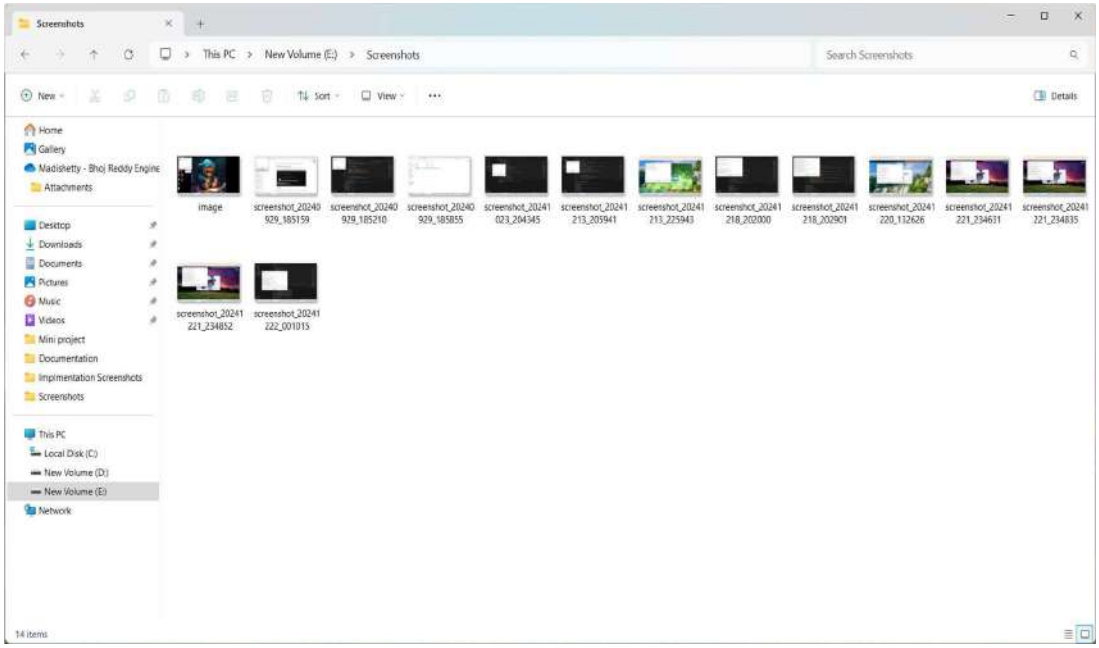

Fig 11 Take Screenshot Output

Input:  or  Capture a Photo

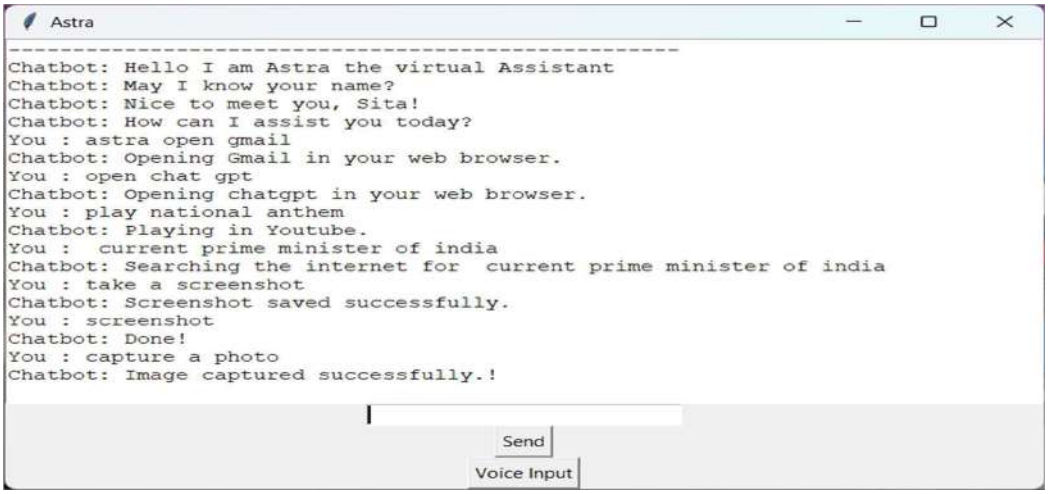

Fig 12 Capture Photo Input

Output: Photo is stored in the Respective file

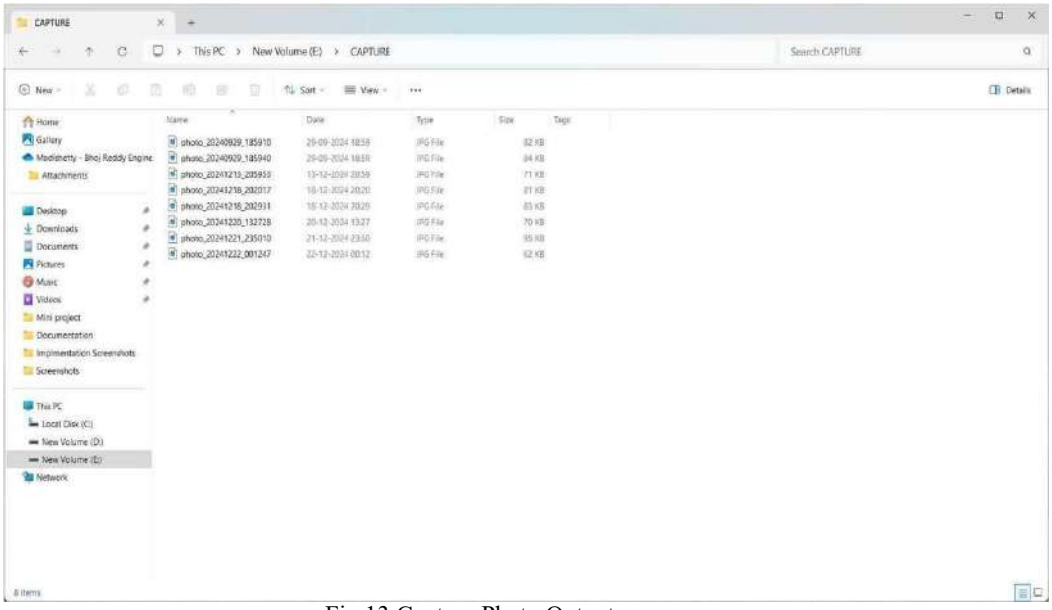

Fig 13 Capture Photo Output

Input:  or  Vathavarnam today

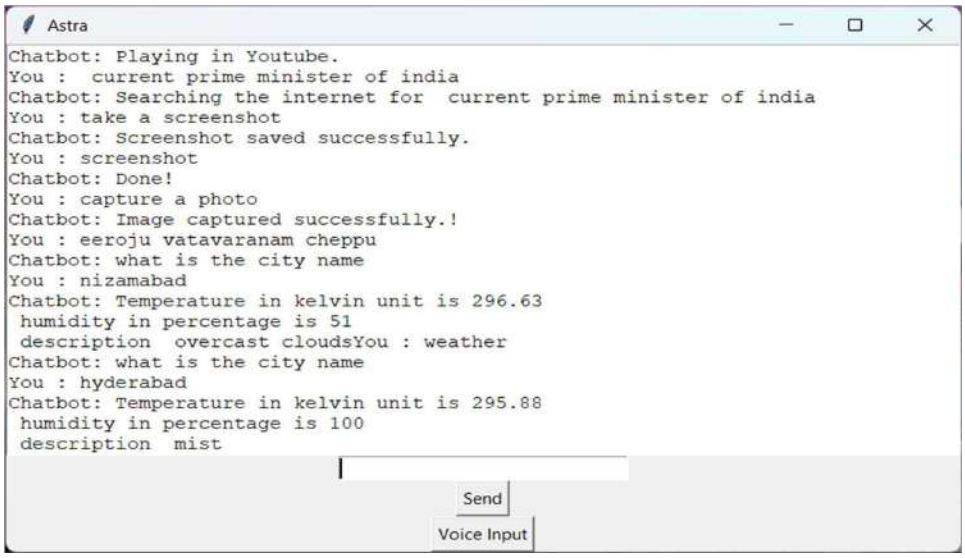

Fig 14 Vatavaranam Result

Input:  or  Show chat history. (It asks for the date in specific format)

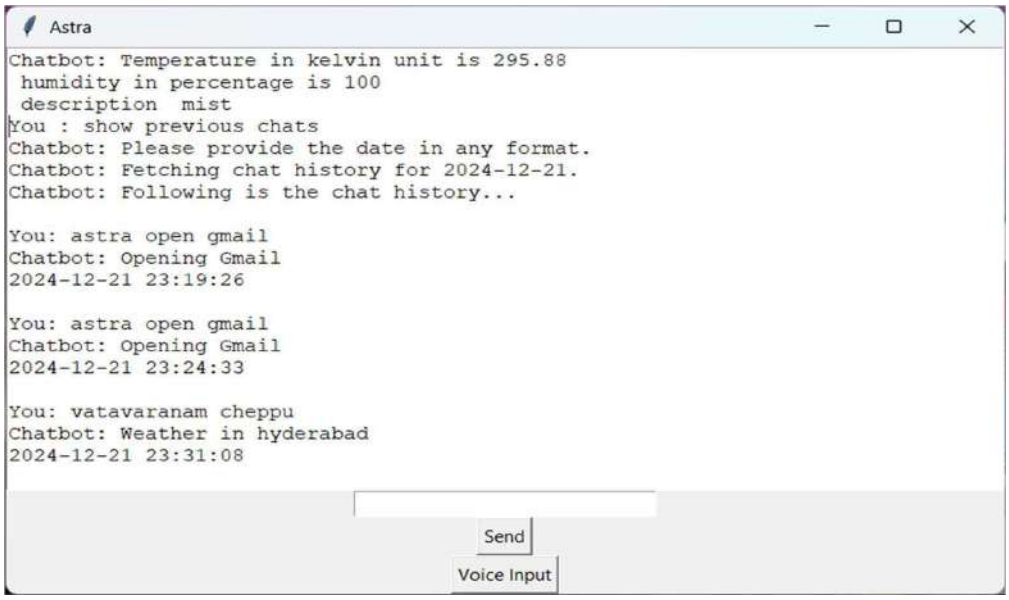

Fig 15 Chat History Result1

Input:  or  Show chat history. (It asks for the date in specific format)

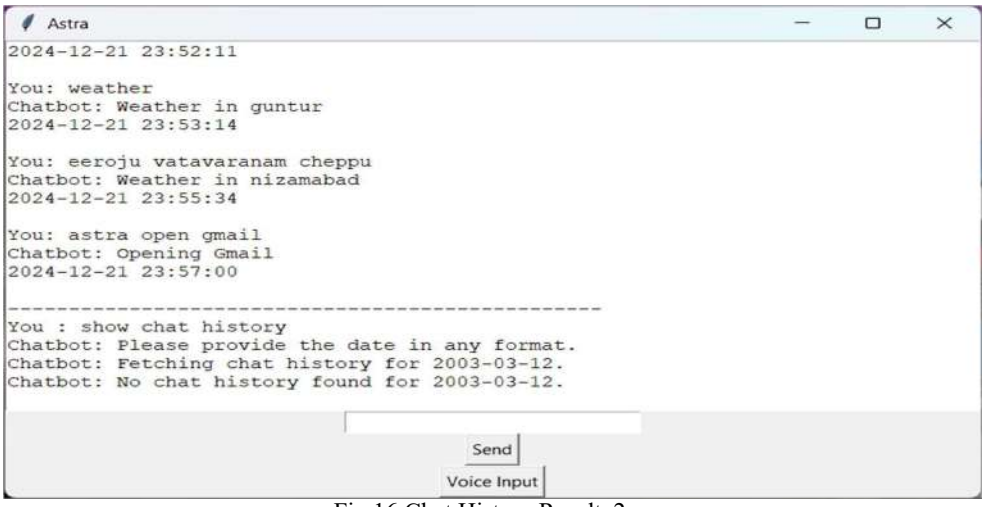

Fig 16 Chat History Result 2

Input:  or  What is my name (It voices out our name which was mentioned at the first meet.)

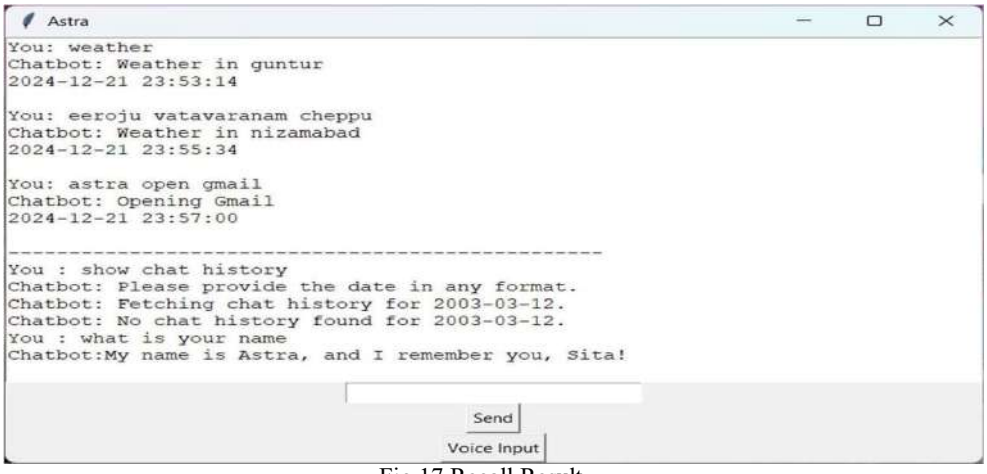

Fig 17 Recall Result

Input:  or  (No input is given by clicking voice input

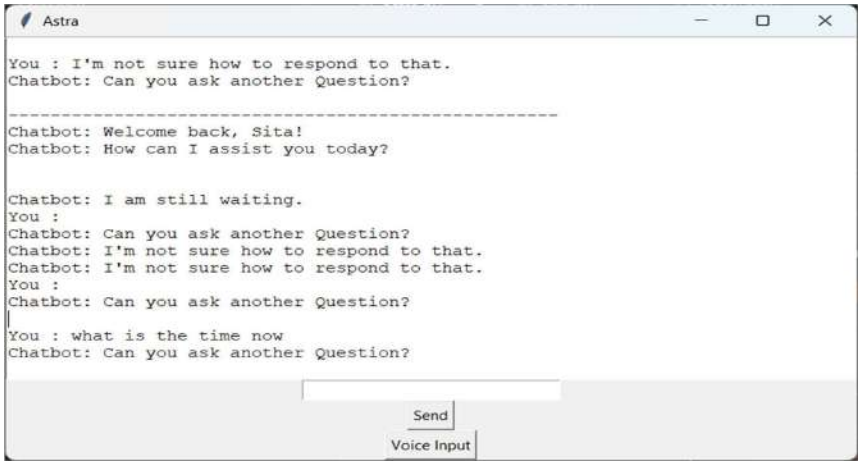

Fig 18 No Input

5. CONCLUSION AND FUTURE SCOPE

ASTRA represents a significant advancement in personal assistant technology for personal computers. By combining robust functionalities with a user-friendly interface, ASTRA promises to enhance productivity, accessibility, and convenience for users, making it an valuable addition to the personal computing environment.

Future Scope

ASTRA Natural Language Understanding (NLU) can be enhanced in the future using sophisticated transformer models. It can be extended as voice-only mode including voice detection. ASTRA can also be created as an offline, stand-alone application.

6. REFERENCES

[1]Comparative Analysis of Smart Voice Assistants. Published in: 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). By R Sivapriyan, N Sakshi, T Vishnu Priya.

[2] OpenAI-Enhanced Personal Desktop Assistant: A Revolution in Human-Computer Interaction. Published in: 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE). By Rijo Benny, Ashwin Muralidharan, Malathi Subramanian.

[3] JARVIS: A PC Voice Assistant Jash Vora1 , Deepak Yadav2 , Ronak Jain3 & Jaya Gupta4 Department of Computer Engineering, A. P. Shah Institute of Technology, G.B.Road, Kasarvadavli, Thane (W), Mumbai-400615, University of Mumbai.