International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 8 Issue 6, Nov-Dec 2024 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

AI Voice Commands Assistant (ZEUS)

Saurabh Darne[1], Pritesh Ingle[2], Samir Satpute[3], Rajneesh Prasad[4], Prof. Shreya Bhanse[5]

[1,2,3,4]School of Sciences, G H Raisoni University, Amravati, Maharashtra, India
[5]Assistant Professor, G H Raisoni University, Amravati, Maharashtra, India

ABSTRACT

As we know Python is an emerging language, so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies is such that they can perform any task with same effectiveness or can say more effectively than us. By making this project It, 1 realized that the concept of Al in every field is decreasing human effort and saving time. It can send emails. It can read PDF It can send text on WhatsApp. command prompt, your favorite IDE, notepad etc. can play music. It can open Now the basic question arises in mind that how it is an Al? The virtual assistant that I have created is like if it is not an Al, but it is the output of a bundle of the statement. But fundamentally, the mail purpose of A.I machines is that it can perform human tasks with the same efficiency or even more efficiently than humans. It is a fact that my virtual assistant is not a very good example of A.I., but it is an A.L.

KEYWORDS: Wake word, Commands, AI (Artificial Intelligence), Skills or Actions, Voice Recognition, Smart Home Integration

How to cite this paper: Saurabh Darne | Pritesh Ingle | Samir Satpute | Rajneesh Prasad | Prof. Shreya Bhanse "AI Voice Commands Assistant (ZEUS)" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-8 | Issue-6, December 2024, pp.79-86, URL: www.ijtsrd.com/papers/ijtsrd70556.pdf

IJTSRD70556

I. INTRODUCTION

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time.

As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time.

The functionalities include , It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation.

Tools and technologies used are PyCharm IDE for making this project, and I created all py files in PyCharm. Along with this I used following modules and libraries in my project. Pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the VVIS as it gives a design and interesting look while having the conversation.

II. PRESENT SYSTEM

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner.

As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. These assistants are no less than a human assistant, but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time.

But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.

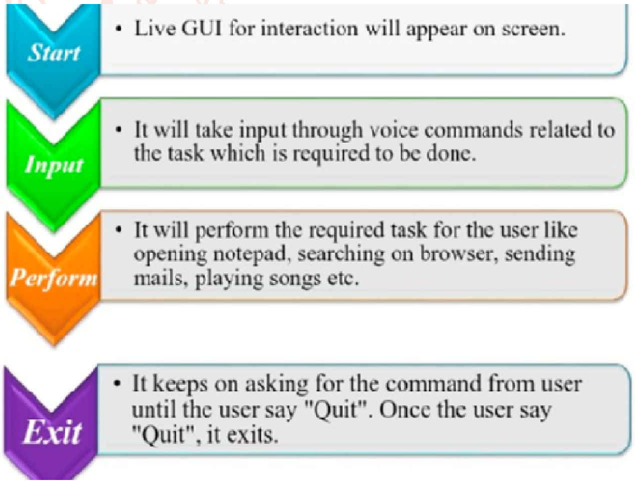
III. PROPOSED SYSTEM

It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. ZEUS is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task. The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the ZEUS as it gives a design and interesting look while having the conversation. With the advancement ZEUS can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation

IV. SYSTEM DESIGN

A. DATA FLOW

The data flow of ZEUS as follows: -



- Live GUI for interaction will appear on screen.
- It will take input through voice commands related to the task which is required to be done.
- It will perform the required task for the user like opening notepad, searching on browser, sending mails, playing songs etc.
- It keeps on asking for the command from user until the user say "Quit". Once the user say "Quit", it exits.

Figure A) Data flow of ZEUS

The system is designed using the concept of Artificial Intelligence and with the help of necessary packages of Python. Python provides many libraries and packages to perform the tasks, for example pyPDF2 can be used to read PDF. The details of these packages are mentioned in Chapter 3 of this report.

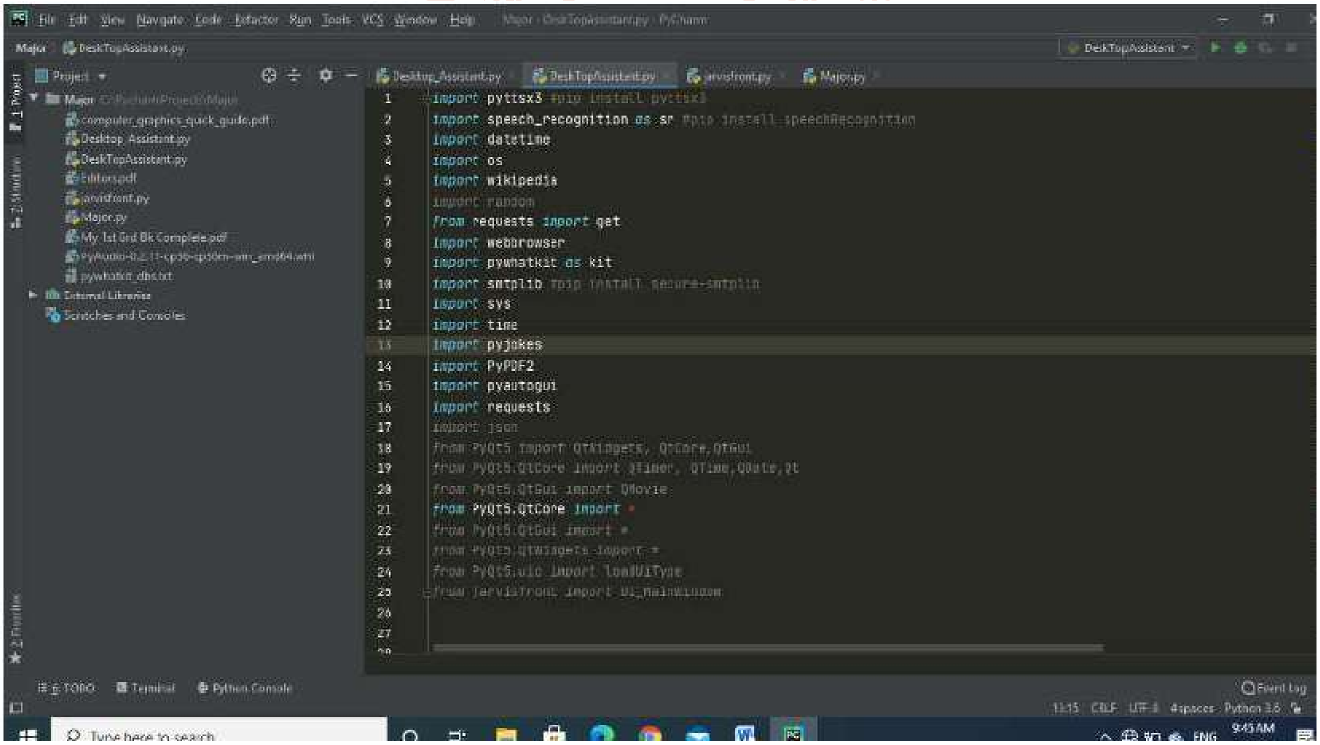
The data in this project is nothing but user input, whatever the user says, the assistant performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e. English. Designing an AI assistant involves a comprehensive architecture that integrates several key components to ensure effective functionality and user experience. At the core, the system architecture consists of a user interface (UI) that supports various interaction modalities, such as voice and text, making it accessible and user-friendly. Natural Language Processing (NLP) is crucial for understanding user queries, utilizing techniques like tokenization and named entity recognition to interpret intent accurately. The backend services handle data storage, typically using SQL or NoSQL databases, and facilitate integration with third-party APIs for functionalities like calendar management and weather updates. Machine learning models, particularly those leveraging deep learning frameworks like TensorFlow or PyTorch, are employed for intent recognition and response generation, continually adapting through user interactions.

Data flows seamlessly from user input through the NLP pipeline, where it is processed and analyzed, to generate context-aware responses. This feedback loop not only enhances user interaction but also contributes to the continuous learning of the system. Security and privacy are paramount; thus, robust mechanisms are implemented to protect user data, ensuring compliance with regulations like GDPR. Furthermore, the design considers potential challenges, such as misinterpretation of commands and user trust, while highlighting opportunities for future enhancements, including the adoption of advanced AI techniques and the integration of emerging technologies like IoT. Ultimately, the goal of this AI assistant system design is to provide a reliable, efficient, and intuitive tool that significantly improves user productivity and interaction quality.

V. SOFTWARE DETAILS

A. PYCHARM

It is an IDE i.e. Integrated Development Environment which has many features like it supports scientific tools(like matplotlib, numpy, scipy etc) web frameworks (example Django,web2py and Flask) refactoring in Python, integrated python debugger, code completion, code and project navigation etc. It also provides Data Science when used with Anaconda



```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import os
import wikipedia
import random
from requests import get
import webbrowser
import pywhatkit as kit
import smtplib #pip install secure-smtplib
import sys
import time
import pyjokes
import PyPDF2
import pyautogui
import requests
import json
from PyQt5 import QtWidgets, QtCore,QtGui
from PyQt5.QtCore import QTimer, QTime,QDate,Qt
from PyQt5.QtGui import QMovie
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
from jarvisfront import Ui_MainWindow
```

Figure A) PyCharm IDE

B. PYQT5 FOR LIVE GUI

PyQt5 is the most important python binding. It contains set of GUI widgets. PyQt5 has some important python modules like QTWidgets, QtCore, QtGui, and QtDesigner etc.

C. PYTHON LIBRARIES

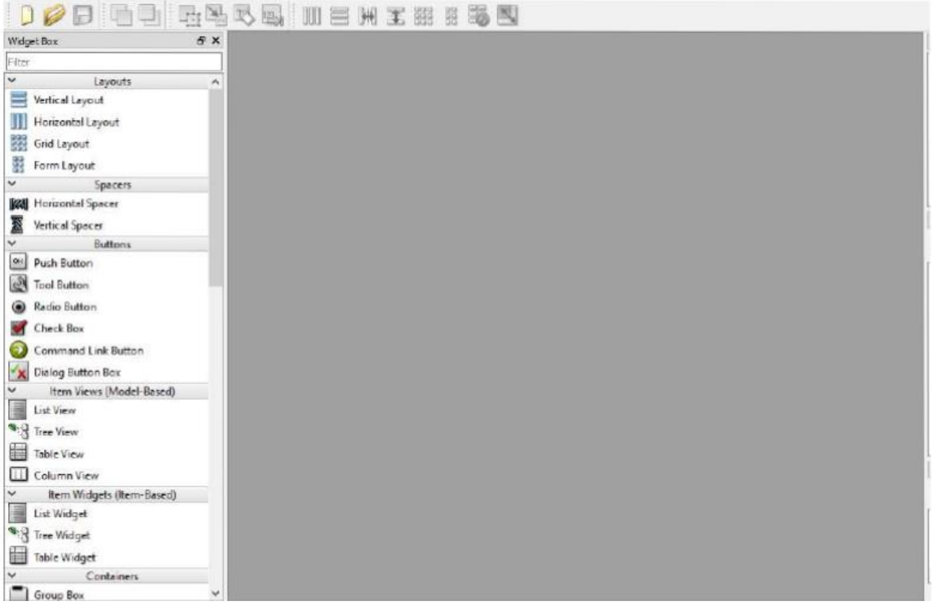


Figure B) PYQT5

In ZEUS following python libraries were used:

a. pyttsx3: It is a python library which converts text to speech.

b. SpeechRecognition: It is a python module which converts speech to text.

c. pywhatkit: It is python library to send WhatsApp message at a particular time with some additional features.

d. Datetime: This library provides us the actual date and time.

e. Wikipedia: It is a python module for searching anything on Wikipedia.

f. Smtplib: Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.

g. pyPDF2: It is a python module which can read, split, merge any PDF.

h. Pyjokes: It is a python libararies which contains lots of interesting jokes in it.

i. Webbrowser: It provides interface for displaying web-based documents to users.

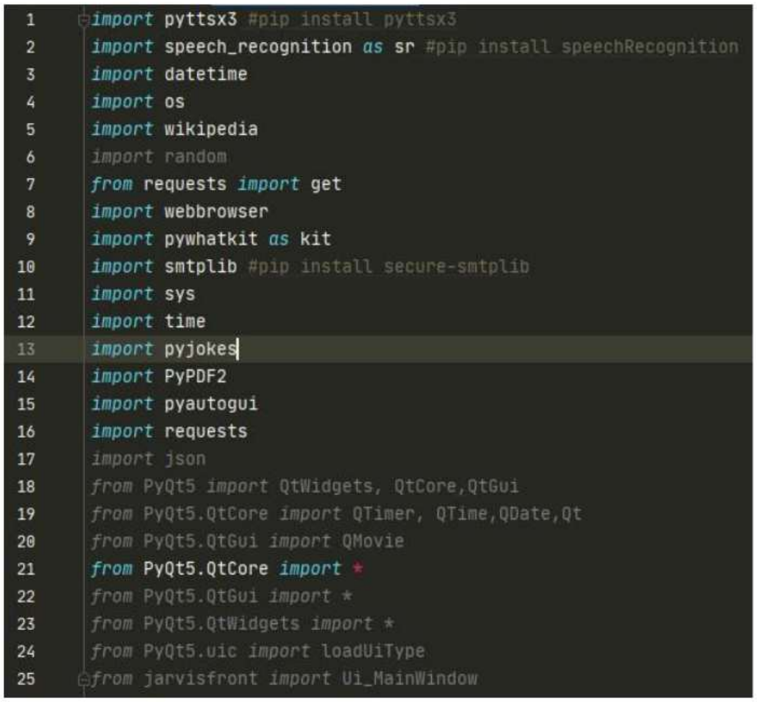
j. Pyautogui: It is a python libraries for graphical user interface.

k. os: It represents Operating System related functionality.

l. sys: It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

Building an AI assistant in Python involves leveraging a variety of libraries that provide essential functionalities ranging from natural language processing to machine learning and web integration. One of the most popular libraries for NLP is spaCy, which offers robust tools for text processing, including tokenization, named entity recognition, and dependency parsing, making it suitable for understanding user inputs. NLTK (Natural Language Toolkit) is another widely-used library that provides a rich set of resources for tasks like stemming, lemmatization, and sentiment analysis. For deep learning tasks, TensorFlow and PyTorch are the go-to frameworks, enabling the development of complex neural networks that can be trained on large datasets for tasks such as intent recognition and response generation.

To facilitate voice interactions, libraries like SpeechRecognition allow the assistant to convert speech to text, while gTTS (Google Text-to-Speech) can convert text responses back into spoken language. For creating conversational interfaces, Rasa is a powerful framework that provides tools for building contextual chatbots using machine learning, enabling the assistant to maintain dialogue state and manage multi-turn conversations effectively. Flask or Django can be used to develop web applications, providing the framework necessary to host the AI assistant and interact with users via web browsers. Additionally, libraries such as requests allow seamless integration with external APIs, enabling the assistant to fetch information from the web, such as weather updates or calendar events. Overall, these libraries collectively empower developers to create sophisticated, responsive, and efficient AI assistants capable of handling a wide range of tasks and interactions.



```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import os
import wikipedia
import random
from requests import get
import webbrowser
import pywhatkit as kit
import smtplib #pip install secure-smtplib
import sys
import time
import pyjokes
import PyPDF2
import pyautogui
import requests
import json
from PyQt5 import QtWidgets, QtCore,QtGui
from PyQt5.QtCore import QTimer, QTime,QDate,Qt
from PyQt5.QtGui import QMovie
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
from jarvisfront import Ui_MainWindow
```

Figure 3.3 Imported Modules

VI. IMPLEMENTATION OF WORK DETAIL

ZEUS, desktop assistant is assistant that can perform many daily tasks of desktop like playing music, opening your favorite IDE with the help of a single voice command ZEUS is different from other traditional voice assistants that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task.

A. REAL LIFE APPLICATION

a. Saves time: ZEUS is a desktop voice assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control and can let us complete set of tasks.

b. Conversational interaction It makes it easier to complete any task as it automatically do it by using the essential module or libraries of Python, in conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done.

c. Reactive nature: The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, ic. Human understandable language, English. So user finds its reaction informed and smart way.

d. Multitasking: The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user "QUIT "it

e. No Trigger phase: It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.

B. DATA IMPLEMENTATION AND PROGRAM EXECUTION

As the first step, install all the necessary packages and libraries. The command used. To install the libraries is "pip install" and then import it. The necessary packages included. Are as follows

a. LIBRARIES AND PACKAGES

b. pytisx3: It is a python library which converts text to speech.

c. SpeechRecognition: It is a python module which converts speech to Text

d. pywhatkit: It is python library to send WhatsApp message at a particular time with some additional features.

e. Datetime: This library provides us the actual date and time.4.2.2.5. Wikipedia: It is a python module for searching anything on Wikipedia.

f. Smtplib: Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.

g. pyPDF2: It is a python module which can read, split, merge any PDF.

h. Pyjokes: It is a python libraries which contains lots of interesting jokes in it.

i. Webbrowser: It provides interface for displaying web-based documents to users.

j. Pyautogul: It is a python library for graphical user interface.

k. os: It represents Operating System related functionality,

l. sys: It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

C. FUNCTIONS

a. takeCommand(): The function is used to take the command as input through microphone of user and returns the output as string.

b. wishMe(): This function greets the user according to the time like Good Morning, Good Afternoon and Good Evening.

c. taskExecution(): This is the function which contains all the necessary task execution definition like sendEmail(), pdf reader(), news() and many conditions in if condition like "open google", "open notepad". "search on Wikipedia", "play music" and "open command prompt" etc.

VII. SYSTEM TESTING

The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for ZEUS desktop assistant focuses on the following four parameters:

A. FUNCTIONALITY

Information Retrieval: Answering questions and providing information on a wide range of topics. Scheduling: Managing calendars, setting reminders, and organizing appointments. Task Management: Helping users create to-do lists and prioritize tasks. Communication: Sending messages, making calls, or drafting emails. Recommendations: Suggesting products, services, or content based on user preferences. Language Translation: Translating text between different languages. Data Analysis: Assisting in analyzing data and generating insights. Learning and Adaptation: Learning integration. From user interactions to provide more personalized responses. Entertainment: Offering games, trivia, or interactive storytelling. The specific capabilities can vary based on the platform and

B. USABILITY

Usability of a system is checked by measuring the easiness of the software and how user friendly it is for the user to use, how it responses to catch query that is being asked by the user.

It makes it easier to complete any task as it automatically do it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done.

The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e. human understandable language, English. So user finds its reaction in an informed and smart way.

The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user "QUIT" it. It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.

C. SECURITY

The security testing mainly focuses on vulnerabilities and risks. As ZEUS is a local desktop application, hence there is no risk of data breaching through remote access. The software is dedicated to a specific system so when the user logs in, it will be activated.

D. STABILITY

Stability of a system depends upon the output of the system, if the output is bounded and specific to the bounded input then the system is said to be stable. If the system works on all the poles of functionality then it is stable. System testing for an AI assistant is a critical phase in the development process, aimed at ensuring the system functions correctly and meets user expectations across various scenarios. This testing encompasses several dimensions, starting with functional testing, which evaluates whether the assistant accurately understands and responds to user inputs. Test cases are designed to cover a wide range of interactions, including simple queries, complex commands, and multi-turn conversations, ensuring that the assistant can handle various linguistic nuances and contextual shifts. Performance testing is also essential, assessing the system's responsiveness and scalability under different loads. This includes simulating numerous simultaneous users to evaluate how well the assistant maintains performance without delays or crashes. Additionally, usability testing involves gathering feedback from real users to identify areas for improvement in the interface and interaction flow, ensuring the assistant is intuitive and user-friendly. Security testing is crucial to safeguard user data and ensure compliance with privacy regulations; this includes testing for vulnerabilities and validating that data encryption and storage protocols are effective. Finally, regression testing ensures that new updates or features do not introduce bugs or adversely affect existing functionalities. By employing a comprehensive testing strategy that encompasses these various aspects, developers can ensure that the AI assistant is robust, reliable, and ready for deployment, ultimately leading to a more satisfying user experience

VIII. CONCLUSION

ZEUS is a very helpful voice assistant without any doubt as it saves time of the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future which are mentioned below:

A. LIMITATIONS

a. Security is somewhere an issue, there is no voice command encryption in this project.

b. Background voice can interfere

c. Misinterpretation because of accents and may cause inaccurate results.

d. ZEUS cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, "Ok Google!"

B. SCOPE FOR FUTURE WORK

a. Make ZEUS to learn more on its own and develop a new skill in it..

b. ZEUS android app can also be developed.

c. Make more ZEUS voice terminals.

d. Voice commands can be encrypted to maintain security.

C. AI ASSISTANTS OFFER SEVERAL ADVANTAGES:

a. Efficiency: They can handle repetitive tasks quickly, saving time for users.

b. 24/7 Availability: Unlike human assistants, AI can work around the clock, providing support anytime.

c. Personalization: AI assistants can learn user preferences and tailor responses and recommendations accordingly.

d. Multitasking: They can manage multiple tasks simultaneously, such as scheduling appointments while providing information.

e. Cost-Effective: Using AI can reduce the need for human resources in certain tasks, leading to cost savings .

f. Data Handling: AI can process and analyze large volumes of data rapidly, offering insights that might be missed otherwise.

g. Accessibility: They can assist users with disabilities by providing voice control and other adaptive features.

h. Consistency: AI assistants deliver consistent performance without the variability that can come from human factors.

i. Integration: They can integrate with various tools and platforms, streamlining workflows and enhancing productivity.

In conclusion, the development of AI assistants represents a significant advancement in human-computer interaction, combining cutting-edge technologies to enhance productivity and streamline daily tasks. These systems leverage a rich ecosystem of libraries and frameworks, enabling them to understand natural language, learn from user interactions, and provide contextually relevant responses. By integrating various functionalities—such as voice recognition, task management, and data retrieval—AI assistants are not only capable of performing routine activities but also of adapting to user preferences, creating a more personalized experience. The challenges associated with natural language understanding and user trust are being progressively addressed through continuous improvements in machine learning algorithms and robust security measures. As AI technology evolves, so too will the capabilities of these assistants, allowing them to engage in more complex conversations and support a wider array of applications across different sectors. Ultimately, the potential for AI assistants to transform how individuals and organizations operate is immense, paving the way for smarter, more efficient workflows that empower users to focus on higher-order tasks and decision-making.

IX. REFERENCES

[1] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "An Analytical Perspective on Various Deep Learning Techniques for Deepfake Detection", *1st International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA),* 10th & 11th June 2022, 2456-3463, Volume 7, PP. 25-30, https://doi.org/10.46335/IJIES.2022.7.8.5

[2] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "Revealing and Classification of Deepfakes Videos Images using a Customize Convolution Neural Network Model", *International Conference on Machine Learning and Data Engineering (ICMLDE)*, 7th & 8th September 2022, 2636-2652, Volume 218, PP. 2636-2652, https://doi.org/10.1016/j.procs.2023.01.237

[3] Usha Kosarkar, Gopal Sakarkar (2023), "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", *4th International Conference on Electrical and Electronics Engineering (ICEEE)*,19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19

[4] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2021), "Deepfakes, a threat to society", *International Journal of Scientific Research in Science and Technology (IJSRST)*, 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682

[5] Usha Kosarkar, Gopal Sakarkar (2024), "Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", Journal of Multimedia Tools and Applications, 1380-7501, https://doi.org/10.1007/s11042-024-19220-w

[6] Usha Kosarkar, Dipali Bhende, "Employing Artificial Intelligence Techniques in Mental Health Diagnostic Expert System", International Journal of Computer Engineering (IOSR-JCE),2278-0661, PP-40-45, https://www.iosrjournals.org/iosr-jce/papers/conf.15013/Volume%202/9.%2040-45.pdf?id=7557