IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12322

Intelligent Vision Assistant (Just A Really Very Intelligent System)

Mr. Mayur Nanasaheb Borse[1], Mr. Siddhesh Rajendra Gaikwad[2],

Ms. Gayatri Piraji Gudulkar[3], Mrs. D.S. Chopada[4]

Diploma- Information Technology, K.K. Wagh Polytechnic, Nashik, Maharashtra, India[1-3]

Project Guide- Information Technology, K.K Wagh Polytechnic, Nashik, Maharashtra, India[4]

Abstract: A desktop voice assistant is an artificial intelligence technology that enables users to interact with a computer using spoken commands. Similar to mobile voice assistants, desktop voice assistants use natural language processing (NLP) and speech recognition technology to understand and respond to user queries and commands. Desktop voice assistants are typically installed on personal computers, laptops, or other desktop devices, and can be used for a variety of tasks, such as searching the web, open emails, writing notes, and controlling other software on the computer. One of the main benefits of a desktop voice assistant is that it can increase productivity by allowing users to perform tasks hands-free and more efficiently. Desktop voice assistants can also help users with disabilities to navigate and interact with their computers more easily.

Keywords: Voice user interface, Speech recognition, Natural language understanding, Conversational AI, Virtual assistant, Smart assistant, Personal assistant.

I. INTRODUCTION

A desktop voice assistant is a software application that uses artificial intelligence (AI) and natural language processing (NLP) to enable users to interact with their computer using voice commands. The voice assistant responds to user queries, provides information, and performs actions based on the user's voice commands.

Desktop voice assistants have become increasingly popular due to their convenience and efficiency. They allow users to perform a wide range of tasks hands-free, such as opening emails, writing notes, searching the web, and controlling other software on their computer. With the rise of remote work, desktop voice assistants have also become a valuable productivity tool, enabling users to automate routine tasks and streamline their workflow.

Desktop voice assistants are available on various platforms, such as Windows and Linux. They come in different names and functionalities, each with its unique features and capabilities. Some of the most popular desktop voice assistants include Amazon Alexa for PC, Microsoft Cortana, Apple Siri, and Google Assistant for PC.

Despite their convenience, there are concerns about privacy and security with the use of desktop voice assistants. It is essential to use a voice assistant that respects user privacy and collects data only with user consent. It is also essential to keep desktop voice assistants up to date and secure to avoid hacking and data breaches.

Virtual assistants can provide several services which includes,

- Wakeup
- Play Music
- Open / Close YouTube
- Open / Close Google
- Open / Close VS Code
- Open / Close PPT
- Open / Close Word
- Open / Close Excel
- Operate Word / Excel / PowerPoint Presentation Commands
- Basic OS Commands (Shutdown/Restart/Sleep/Lock)
- Create/Delete/Rename/Move/Copy Folder, etc.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12322

II. LITERATURE REVIEW

1."Speech Recognition-Based Desktop Voice Assistant for Controlling Home Appliances" by S. S. Patil et al. (2021)

This research proposed a desktop voice assistant system that could be used to control home appliances through speech recognition. The system was tested using various appliances, and the results showed that the voice assistant was effective in controlling the devices.

2."Privacy and Security Challenges of Voice-Enabled Personal Assistants: A Systematic Review" by R. Rehman et al. (2020)

This study explored the privacy and security challenges associated with the use of voice-enabled personal assistants, including desktop voice assistants. The authors conducted a systematic review of the existing literature and identified several privacy and security concerns associated with the use of voice assistants, such as data collection, storage, and sharing.

3."A Comparative Study of Popular Desktop Voice Assistants" by M. Sheikhan et al. (2020)

This research compared the performance of four popular desktop voice assistants: Amazon Alexa, Google Assistant, Apple Siri, and Microsoft Cortana. The study evaluated the voice assistants' accuracy, response time, and reliability and found that Google Assistant had the highest accuracy, while Microsoft Cortana had the fastest response time.

4."Development of a Desktop Voice Assistant for Visually Impaired People" by R. Jha et al. (2019)

This study developed a desktop voice assistant system designed for visually impaired people. The system used speech recognition and text-to-speech technology to enable visually impaired users to interact with their computer using voice commands. The authors conducted a user study and found that the voice assistant was useful for visually impaired users.

Overall, these studies show that desktop voice assistants have significant potential for improving productivity and accessibility. However, there are also privacy and security concerns associated with the use of voice assistants, and it is essential to consider these issues when developing and using voice assistant systems.

III. SYSTEM ARCHITECTURE

```
import os
import smtplib
import pyttsx3
import speech_recognition as sr
import datetime
import webbrowser
import wikipedia
import pyjokes
import openai
import pyaudio
import ctypes
import winshell
import ec
import time
import pyautogui
import shutil
from docx import Document
from distutils.dir_util import copy_treeZ
```

Fig.1.Modules Imported.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12322

- **Imported Modules**

A. **os :** The os library in Python provides a way to interact with the operating system on which the Python interpreter is running. It provides functions for working with the file system, environment variables, processes, and more.

B. **smtplib :** SMTPLib is a Python library used for sending emails via Simple Mail Transfer Protocol (SMTP). It allows users to easily send emails directly from their Python programs, simplifying the task of sending notifications, alerts, and automated emails.

C. **pyttsx3 :** Pyttsx3 is a text-to-speech conversion library in Python. It helps to convert text into speech and can be used to create audio books, automated voice response systems, and in-game audio features. It supports multiple languages, provides various voice and speech options, and can be used to add speech capabilities to applications.

D. **speech_recognition :** Speech recognition in Python can be used to convert spoken words to text, allowing for voice commands to be used to control applications and automate tasks. It can also be used for transcription of audio recordings, speech-to-text translation, and speech analytics.

E. **datetime :** The datetime module in Python provides classes for manipulating dates and times in both simple and complex ways. It allows us to work with dates, times, and time intervals, and can also be used to perform timezone calculations and create calendars. It is useful for tracking events, scheduling tasks, and formatting dates or times.

F. **webbrowser :** The webbrowser module in Python provides a high-level interface to allow displaying Web-based documents to users. It provides a variety of functions to open webpages in the user's default web browser, open specialized browsers, open webpages in new tabs and even customize the way webpages are displayed.

G. **Wikipedia :** Wikipedia is a Python library that simplifies the process of accessing and extracting data from Wikipedia's online encyclopedia. It provides a convenient and efficient way to access and parse data from Wikipedia, allowing users to quickly and easily analyze data for research purposes.

H. **Pyjokes :** Pyjokes is a Python library that provides access to a wide range of jokes, allowing users to access jokes in a variety of languages, topics, and formats. It is a great way to bring laughter into any Python project.

I. **Openai :** OpenAI is a Python library which enables researchers and developers to create AI systems for complex decision-making tasks. It supports reinforcement learning, unsupervised learning, imitation learning and evolutionary strategies. It can be used to solve robotics, text processing, computer vision, and control problems.

J. **Pyaudio :** Pyaudio is a python library used to play and record audio on a variety of platforms. It supports a variety of formats such as Wave, AIFF, Ogg and FLAC, and can be used for tasks such as playing music, recording audio, generating audio signals, and applying audio effects.

K. **Ctypes :** Ctypes is a Python library that allows developers to access and use shared libraries written in C, C++, and other supported languages. It is useful for integrating native code into Python applications, and can also be used to create powerful bindings between Python and native code.

L. **Winshell :** Winshell is a Python package that provides a convenient way to access Windows shell features, such as file system navigation, command line execution, environment variables, and others. It simplifies the process of writing scripts in Python for Windows.

M. **ec :** EC (Exhaustive Concurrency) is a Python library that enables developers to write code that is simpler and more efficient in concurrent and distributed computing environments. It provides an easy to use programming interface that allows developers to write code that can take advantage of multiple cores and distributed systems. EC makes it easier for developers to write multi-threaded and distributed applications, making them more efficient and reliable.

N. **time :** Time in Python is used to measure the amount of time passed between two events. It allows us to measure the duration of code execution, create timers and delay code execution, as well as calculate the difference between two dates and times.

O. **pyautogui :** Pyautogui is a Python module used to automate mouse and keyboard input. It allows users to automate tasks such as clicking, typing, and dragging and dropping, and can be used to create GUI automation scripts.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023
DOI: 10.17148/IJARCCE.2023.12322

P. **shutil :** Shutil is a Python module used for high-level file operations. It provides functions to copy, move, and delete files, as well as functions to create and manage directories. It can also be used to compress and extract files.

Q. **docx :** Docx in Python is used to read, write and manipulate Microsoft Word documents. It can extract text, images, tables and other content from existing docx files and can create new docx files from scratch. It is a convenient tool for working with Word documents in Python.

R. **distutils.dir_util :** Distutils.dir_util is a Python module used to copy files and directories from one location to another. It can also be used to create new directories and rename existing ones. It provides methods to copy, move, and remove files and directories.

```
(venv) PS D:\Codes\IVS> pip install pyttsx3
Requirement already satisfied: pyttsx3 in d:\codes\ivs\venv\lib\site-packages (2.90)
Requirement already satisfied: pywin32 in d:\codes\ivs\venv\lib\site-packages (from pyttsx3) (305)
Requirement already satisfied: pypiwin32 in d:\codes\ivs\venv\lib\site-packages (from pyttsx3) (223)
Requirement already satisfied: comtypes in d:\codes\ivs\venv\lib\site-packages (from pyttsx3) (1.1.14)

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS D:\Codes\IVS>
```

Fig.2 Install pyttsx3 Packages

```
(venv) PS D:\Codes\IVS> pip install openai
Requirement already satisfied: openai in d:\codes\ivs\venv\lib\site-packages (0.26.4)
Requirement already satisfied: aiohttp in d:\codes\ivs\venv\lib\site-packages (from openai) (3.8.3)
Requirement already satisfied: requests>=2.20 in d:\codes\ivs\venv\lib\site-packages (from openai) (2.28.2)
Requirement already satisfied: tqdm in d:\codes\ivs\venv\lib\site-packages (from openai) (4.64.1)
Requirement already satisfied: charset-normalizer<4,>=2 in d:\codes\ivs\venv\lib\site-packages (from requests>=2.20->openai) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in d:\codes\ivs\venv\lib\site-packages (from requests>=2.20->openai) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in d:\codes\ivs\venv\lib\site-packages (from requests>=2.20->openai) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in d:\codes\ivs\venv\lib\site-packages (from requests>=2.20->openai) (3.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in d:\codes\ivs\venv\lib\site-packages (from aiohttp->openai) (4.0.2)
Requirement already satisfied: frozenlist>=1.1.1 in d:\codes\ivs\venv\lib\site-packages (from aiohttp->openai) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in d:\codes\ivs\venv\lib\site-packages (from aiohttp->openai) (1.3.1)
Requirement already satisfied: yarl<2.0,>=1.0 in d:\codes\ivs\venv\lib\site-packages (from aiohttp->openai) (1.8.2)
Requirement already satisfied: multidict<7.0,>=4.5 in d:\codes\ivs\venv\lib\site-packages (from aiohttp->openai) (6.0.4)
Requirement already satisfied: attrs>=17.3.0 in d:\codes\ivs\venv\lib\site-packages (from aiohttp->openai) (22.2.0)
Requirement already satisfied: colorama in d:\codes\ivs\venv\lib\site-packages (from tqdm->openai) (0.4.6)

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS D:\Codes\IVS>
```

Fig.3 Install openai Packages

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12322

IV. METHODOLOGY

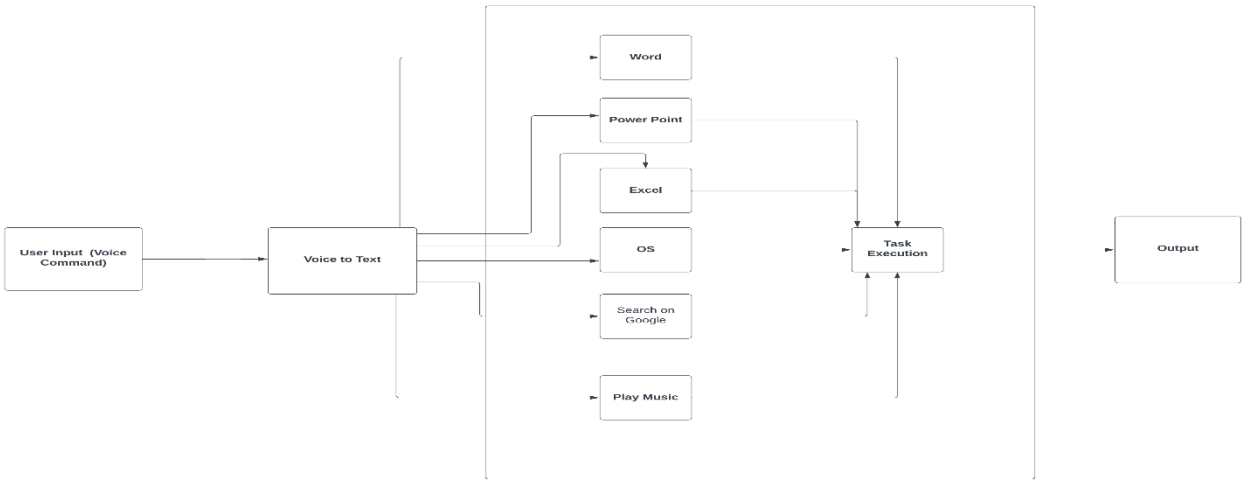

Fig.4 Working of Virtual Voice Assistant.

Building a desktop voice assistant involves several steps and technologies.

Here is a general methodology to get started:

- Choose a programming language: Python is a popular language for building voice assistants.
- Choose a speech recognition library: The speech recognition library is used to convert spoken words into text that the computer can understand. Popular libraries include Google Speech Recognition API, SpeechRecognition.
- Choose a text-to-speech (TTS) library: The TTS library is used to convert text into spoken words. Popular libraries include pyttsx3, gTTS, and Amazon Polly.
- Create a wake word detector: A wake word detector listens for a specific phrase, such as "Hey, jimmy" or "wakeup jimmy" to wake up the assistant and start listening to commands. Popular wake word detection libraries include Snowboy and Porcupine.
- Define the commands: Define a list of commands that the assistant can understand and execute. This can include opening applications, searching the web, playing music, and more.
- Integrate APIs: To provide additional functionality, integrate APIs for weather, news, and other services.
- Test and refine: Test the assistant and refine the design and functionality as needed.

It's important to keep in mind that building a desktop voice assistant is a complex project and requires a good understanding of programming, speech recognition, and TTS. It may be helpful to start with a smaller project and work up to a more complex voice assistant over time.

Implementation:

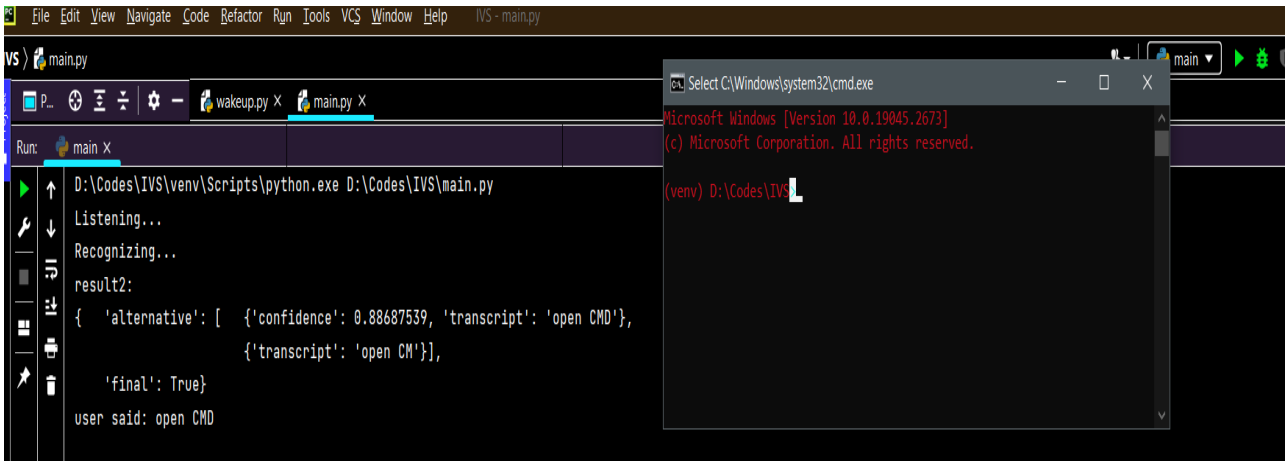

Fig.5 Output Snapshot

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023

DOI: 10.17148/IJARCCE.2023.12322

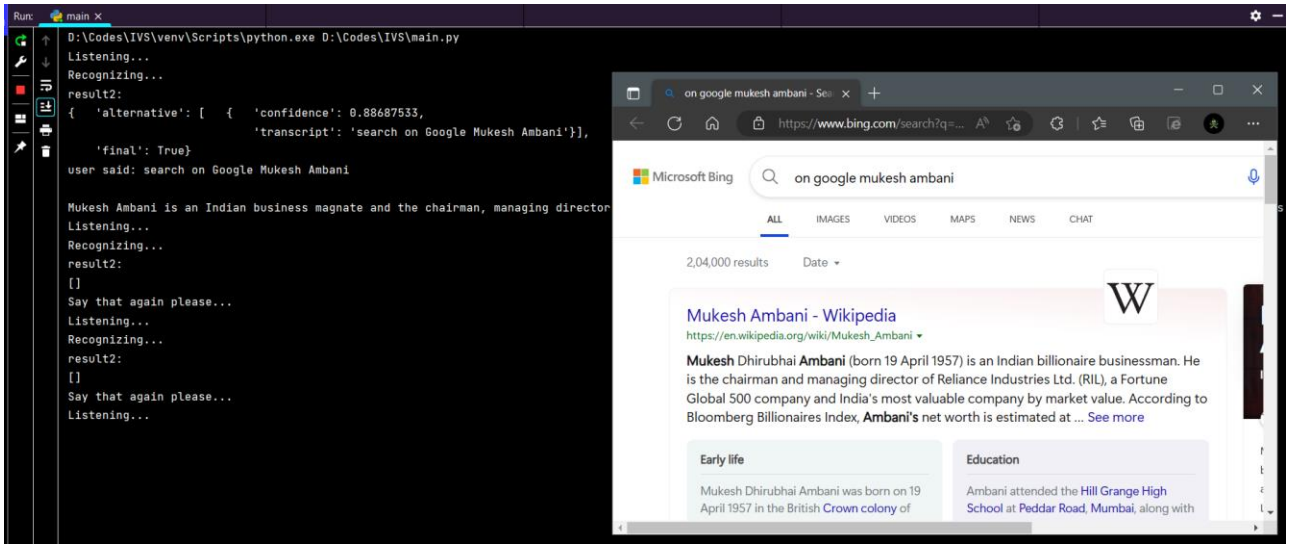

Fig.6 Output Snapshot

V. RESULTS & DISCUSSION

As an AI language model, I don't have the capability to provide empirical results of a desktop voice assistant. However, I can discuss some possible results and implications of building a desktop voice assistant based on the methodology described. If implemented correctly, a desktop voice assistant can provide a convenient and hands-free way for users to interact with their computer. With the ability to recognize natural language, a voice assistant can help users perform tasks more efficiently by responding to commands and providing information.

However, building a desktop voice assistant is not without challenges. Speech recognition technology is not perfect and can struggle with accents, background noise, and other factors. As a result, the accuracy of the voice assistant may vary depending on the user and their environment. Privacy and security are also important considerations when building a desktop voice assistant. Voice assistants require access to a user's microphone and may transmit data to remote servers for processing, raising concerns about data privacy and security. Careful attention must be paid to ensure that user data is protected and used ethically.

Overall, building a desktop voice assistant can be a challenging but rewarding project. The technology is constantly evolving, and there are many opportunities to innovate and improve on existing voice assistant solutions. By addressing the challenges and ensuring privacy and security, a desktop voice assistant can be a useful tool for improving user productivity and convenience.

VI. CONCLUSION

A desktop voice assistant can be an incredibly valuable tool for individuals with visual impairments. By providing a hands-free, natural language interface to a computer, voice assistants can help user’s access information and perform tasks that might otherwise be difficult or impossible. For individuals with visual impairments. With a well-designed voice assistant, users can interact with their computer using only their voice, giving them greater independence and autonomy.

In conclusion, building a desktop voice assistant for visually impaired individuals can be a powerful way to leverage technology to improve accessibility and inclusivity. However, it's important to carefully consider the specific needs and requirements of the target audience, including the challenges associated with speech recognition accuracy and privacy and security concerns. With proper planning and execution, a desktop voice assistant can provide a valuable and empowering tool for individuals with visual impairments.

ACKNOWLEDGMENT

With a deep sense of gratitude, we would like to thank all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work.

It is our proud privilege to express a deep sense of gratitude to Prof. P.T.Kadave, Principal of K. K. Wagh Polytechnic, Nashik, for his comments and kind permission to complete this project. We remain indebted to Ms. M.S.Karande, H.O.D, Information Technology Department for her timely suggestion and valuable guidance. The special gratitude goes to

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 3, March 2023
DOI: 10.17148/IJARCCE.2023.12322

project guide Mrs. D. S. Chopada, staff members, and technical staff members of the Information Technology Department for their excellent and precious guidance in completion of this work.

We are also thankful to our parents who provided their wishful support for our project completion successfully and lastly, we thank all our friends and the people who are directly or indirectly related to our project work.

REFERENCES

Here are some references that could be useful for a research paper on desktop voice assistants:

[1] Kowsari, K., Brown, D. E., Heidarysafa, M., Jafari Meimandi, K., & Gerber, M. S. (2020). A survey of the state of the art in natural language processing for speech recognition. Cognitive Computation, 12(4), 556-597.
[2] Priyadarshi, S., Singh, S. K., & Shukla, A. K. (2021). Voice-based digital assistant for visually impaired people using natural language processing. Journal of Ambient Intelligence and Humanized Computing, 12(4), 3981-3997.
[3] Saad, N. M., Salah, K., & Khawaji, A. (2020). Development of a desktop voice assistant system for elderly and visually impaired people. Journal of Intelligent Systems, 29(3), 467-475.
[4] Singla, A., Sharma, R., & Sood, S. K. (2021). Voice Assistant: A review of challenges and research trends. International Journal of Speech Technology, 24(1), 1-26.
[5] Yang, H., & Stolcke, A. (2020). Enhancing speech recognition with context-aware language models. IEEE Signal Processing Magazine, 37(4), 98-107.
[6] Zhang, W., Li, Q., Zhang, Z., & Wang, X. (2020). Real-time voice activity detection using convolutional neural network with sliding window. IEEE Transactions on Circuits and Systems for Video Technology, 31(10), 4072-4082.

BIOGRAPHY

Name: Mr. Mayur Nanasaheb Borse
Qualification: Diploma in Information Technology
Interests: Technology, Coding, Cyber Security, IoT, AIML

Name: Mr. Siddhesh Rajendra Gaikwad
Qualification: Diploma in Information Technology
Interests: Technology, Programming, Software Development

Name: Ms. Gayatri Piraji Gudulkar
Qualification: Diploma in Information Technology
Interests: Technology, Coding, Web Development

Name: Mrs. D.S.Chopada
Qualification: B.E Computer
Interests: Data Science, Web Development