WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Personal Assistant Using Artificial Intelligence

*Preeti Jagdale, Archish Shetty, Hritwij Rale, Dr. S. A. Shirsat*

Electronics and Telecommunication Department, Sinhgad College of Engineering

**ABSTRACT**

In this project a Personal assistant is build using Artificial Intelligence. The project is build using available open source software modules with visual studio community backing which can accommodate any updates in future. The use of voice based personal assistants is increasing day by day and helping in making our life simple. This project presents intelligent voice assistant with ability to organize and maintain information it includes the management of emails, calendar events, files and to do lists. The work is going on to develop an "Personal Voice Assistant using Python "which will their voice (speech), extract information and perform tasks on their desktop.

**Keywords:** PyTTSx , Wikipedia , Data Time , Calendar ,Smtplib.

## I. INTRODUCTION

The project Personal assistants are an important achievement, which have become an indispensable part of the ubiquitous digitalization process. These virtual assistants can be found in all gadgets such as smartphones, tablets and also smart watches now. The increasing competition in this area has led to many improvements. Big companies like Amazon, Google, Microsoft and Apple offer a complete digital infrastructure that can be controlled by voice assistants. A virtual assistant is a ***technology based on artificial intelligence***. The software uses a device's microphone to receive voice requests while the voice output takes place at the speaker. But the most exciting thing happens between these two actions. It is a combination of several different technologies: voice recognition, voice analysis and language processing. When a user asks a personal assistant to perform a task, the natural language audio signal is converted into digital data that can be analyzed by the software. Then this data is compared with a database of the software using an innovative algorithm to find a suitable answer. This database is located on distributed servers in cloud networks. For this reason, most personal assistants cannot work without a reliable Internet connection. With the increasing number of queries the software's database gets expanded and optimized, which improves voice recognition and increases the response time of the system.

## II. METHODOLOGY

The foremost function is then made public during which all of the competencies of this

technique rectangular degree are made public. The planned system is meant to possess the following functionality:

a) The Dollar asked the user for input and keeps listening for orders. The time for a hearing is visiting be set up in step with the user's control.

b) If the assistant fails to grasp the command it's visiting keep asking the user to repeat the command once again} and yet again.

c) This assistant is visiting be bespoken to possess either male or female voice in step with user's demand.

d) the current version of the assistant supports choices like Checking weather updates, deed and checking emails, Searching Wikipedia, Stream music, Open applications, Text messages, checking dates and times, taking notes, show notes, Open YouTube, etc. 213 International Journal for Modern Trends in Science and Technology Streaming Music: The user can command Dollar to play a music track and it's going to execute a command and search into it from the song Folder. Read the latest news from headlines: Dollar will examine out latest headlines from the knowledge retailers of the required topics you care about or need information Keep Tabs on the traffic & the weather: Dollar can research the weather forecast or alert you if there is an accident that will delay your morning journey. Set Reminders/timers: You'll be able to tell Dollar to wake you up daily morning at 4 a.m.

Answer the following questions: Dollar can look up simple information, solve mathematical problems, or tell you a joke However, Alexa, Sleek, and homey aren't the sole ones altogether of most home automation systems that reply to user voice directions.

Castles has been around since late 2012, however, its huge hub and Kinetic voice management programmer will solely be used on a Windows machine.

## III BLOCK DIAGRAM

### CALENDAR MODULE

Python defines an inbuilt module calendar that handles operations related to the calendar.

The calendar module allows us to output calendars like the program and provides additional useful functions related to the calendar. Functions and classes defined in the calendar module use an idealized calendar, the current Gregorian calendar extended indefinitely in both directions. By default, these calendars have Monday as the first day of the week, and Sunday as the last (the European convention). Python program to print Calendar without calendar or datetime module 2. Calendar Functions in Python | Set 1 (calendar (), month (), isleap ()...) 3. Python calendar module : format month () method 4. Python calendar module | itermonthdays2 () method 5. Python calendar module: iterweekdays.

### WIKIPEDIA MODULE

The Internet is the single largest source of information, and therefore it is important to know how to fetch data from various sources. And with Wikipedia being one of the largest and most popular sources for information on the Internet. Wikipedia is a multilingual online encyclopedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system.

In this article, we will see how to use Python's Wikipedia module to fetch a variety of information from the Wikipedia website**.** Python's Wikipedia module can be used to fetch a bunch of information from the Wikipedia website we are all familiar with. We will be starting off by importing the wikipedia module into our program. If the importing command gives an error. Make sure you install the module using the pip command.

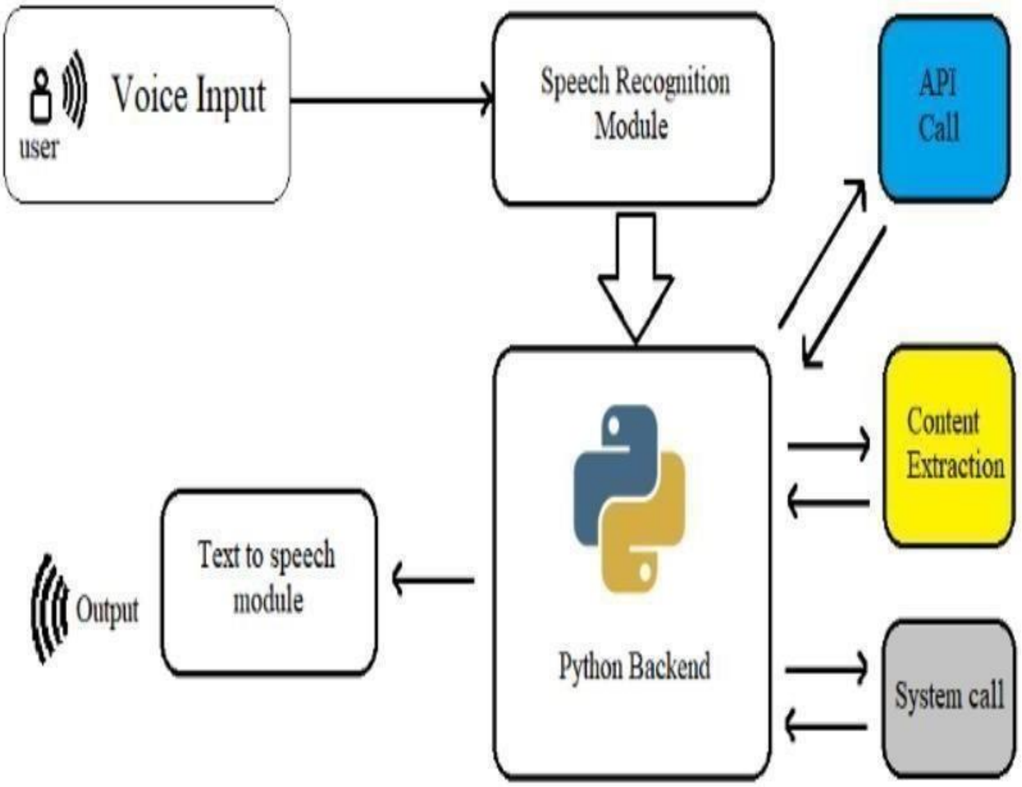


**Fig no 1 Block diagram of Personal Assistant using Artifical Intelligence.**

### PYTTX MODULE

Pyttsx is a cross-platform text to speech library which is platform independent. The major advantage of using this library for text-to-speech conversion is that it works offline. However, pyttsx supports only Python 2.x. Hence, we will see pyttsx3 which is modified to work on both Python 2.x and Python 3.x with the same code. Open a terminal or command prompt and type the following command: Once installed, we can start using Pyttsx3 to build text to speech (TTS) applications in Python. Using Pyttsx3 is straightforward. We first need to import the pyttsx3 library in our Python code.The best thing about pyttsx is that it works offline without any kind of delay. Install it via pip . By default, the pyttsx3 library loads the best driver available in an operating system: nsss on Mac, sapi5 on Windows and espeak on Linux and any other platform. Import the installed pyttsx3 into your program.

### SMTPLIB MODULE

The smtplib is a Python library for sending emails using the Simple Mail Transfer Protocol (SMTP). The smtplib is a built-in module; we do not need to install it. It abstracts away all the complexities of SMTP. To actually send an email, we need to have access to a mail server. Python comes with a simple development mail server.The Port number used here is ‘587’.The smtplib module defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult RFC 821 (Simple Mail Transfer Protocol) and RFC 1869 (SMTP Service Extensions).

### REQUEST MODULE

Requests library is one of the integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scraping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response.

Make requests using a variety of different HTTP methods such as GET, POST, and PUT Customize your requests by modifying headers, authentication, query strings, and message bodies Inspect the data you send to the server and the data the server sends back to you.

### DATA TIME MODULE

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module comes built into Python, so there is no need to install it externally. Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

The DateTime module is categorized into 6 main classes –

date – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.

time – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.

datetime – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.

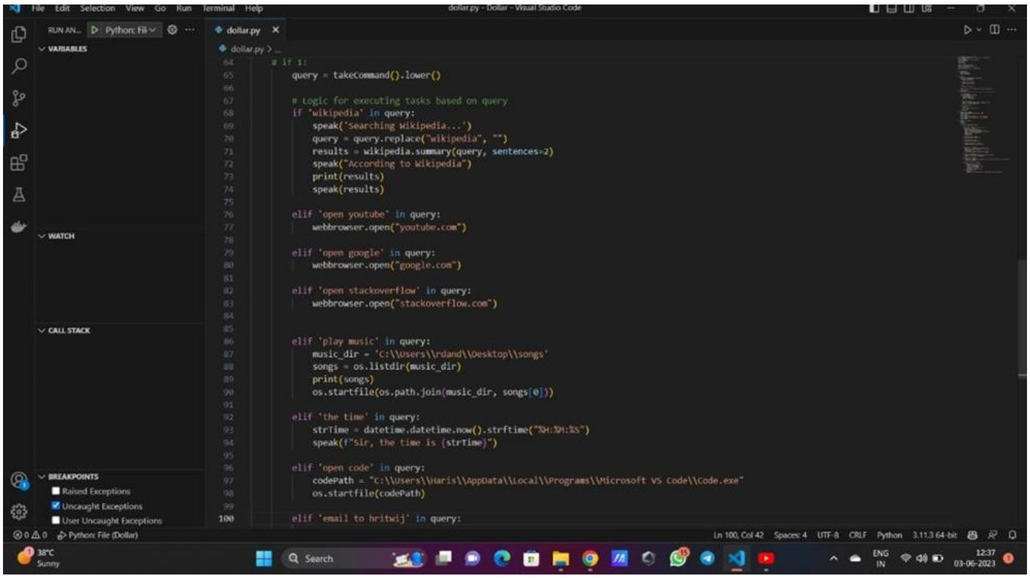
timedelta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

tzinfo – It provides time zone information objects.

timezone – A class that implements the tzinfo abstract base class as a fixed offset from the UTC (New in version 3.2).

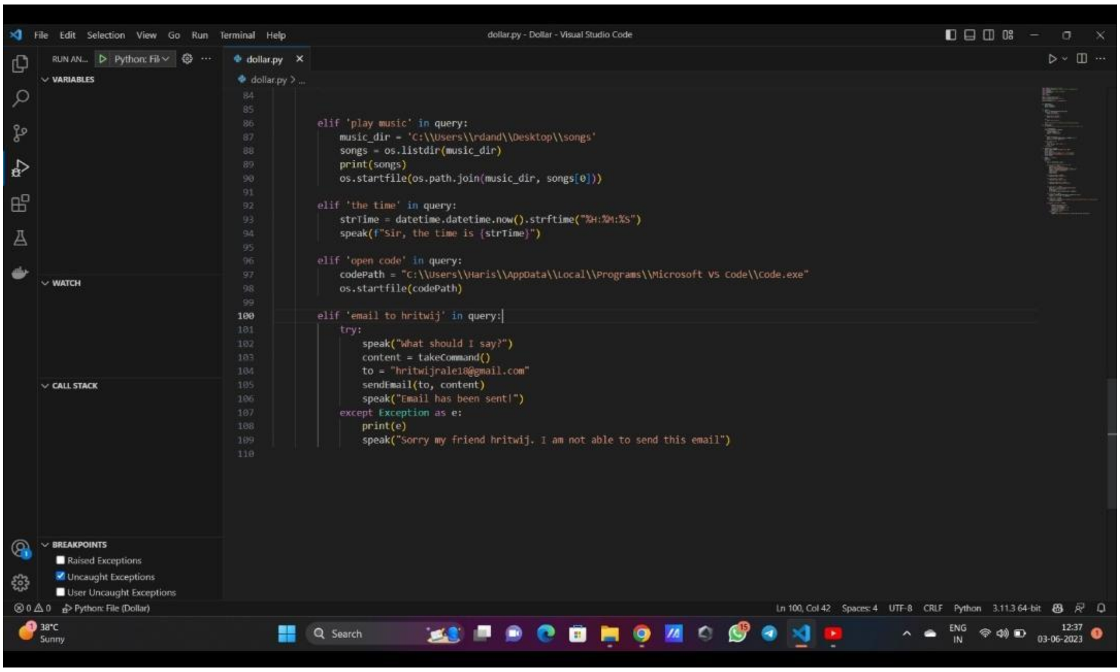
## IV. MODELING AND ANALYSIS

A model analysis is the process of evaluating and interpreting the performance, accuracy, and reliability of a model to assists its effectiveness in solving a specific problem or making predictions.



□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Exceptions

```python
# if 1:
    query = takeCommand().lower()

    # Logic for executing tasks based on query
    if 'wikipedia' in query:
        speak('Searching Wikipedia...')
        query = query.replace("wikipedia", "")
        results = wikipedia.summary(query, sentences=2)
        speak("According to Wikipedia")
        print(results)
        speak(results)

    elif 'open youtube' in query:
        webbrowser.open("youtube.com")

    elif 'open google' in query:
        webbrowser.open("google.com")

    elif 'open stackoverflow' in query:
        webbrowser.open("stackoverflow.com")


    elif 'play music' in query:
        music_dir = 'C:\\Users\\rdand\\Desktop\\songs'
        songs = os.listdir(music_dir)
        print(songs)
        os.startfile(os.path.join(music_dir, songs[0]))

    elif 'the time' in query:
        strTime = datetime.datetime.now().strftime("%H:%M:%S")
        speak(f"Sir, the time is {strTime}")

    elif 'open code' in query:
        codePath = "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
        os.startfile(codePath)

    elif 'email to hritwij' in query:
```


File Edit Selection View Go Run Terminal Help
dollar.py - Dollar - Visual Studio Code
RUN AN... Python: Fil
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Exceptions
dollar.py
Ln 100, Col 42 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
38°C Sunny
Search
12:37 03-06-2023

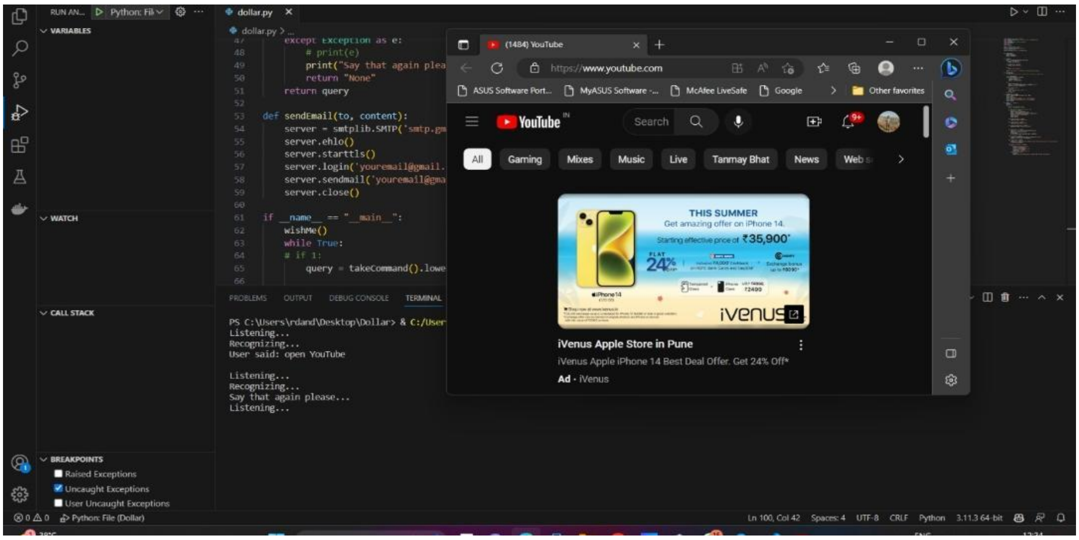
```python


        elif 'play music' in query:
            music_dir = 'C:\\Users\\rdand\\Desktop\\songs'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[0]))

        elif 'the time' in query:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"Sir, the time is {strTime}")

        elif 'open code' in query:
            codePath = "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
            os.startfile(codePath)

        elif 'email to hritwij' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "hritwijrale18@gmail.com"
                sendEmail(to, content)
                speak("Email has been sent!")
            except Exception as e:
                print(e)
                speak("Sorry my friend hritwij. I am not able to send this email")

```

**Fig no 2: Image of code analyzing**

## V. RESULTS AND DISCUSSION

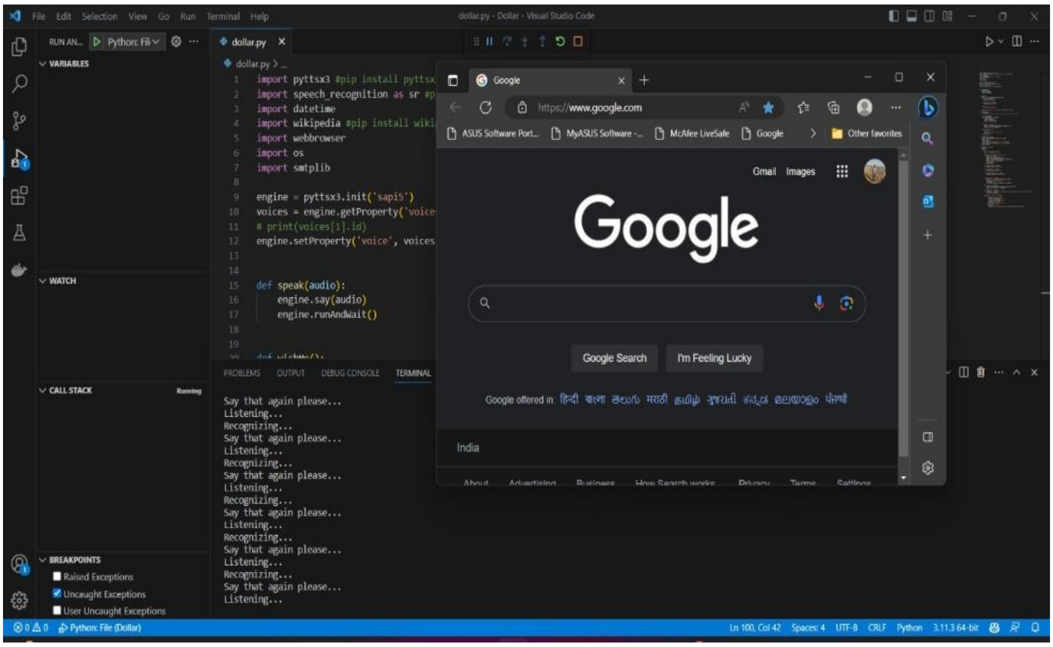
This section describes a brief description of our result on the basis of the comparison and analysis of our proposed work. We have employed this idea by means of Python, Machine Learning and AI. Our main aim is to assist the users in their tasks with the help of their voice commands. This can be done in two phases. Firstly, taking the audio input from the user and converting it to an English phrase with the help of Speech Recognition API.

Secondly searching for the task user wants to perform and then redirecting it to the Linux server with the help of HTTP Protocol and displaying the result on the web browser.



```python
        except Exception as e:
            # print(e)
            print("Say that again plea
            return "None"
        return query

def sendEmail(to, content):
        server = smtplib.SMTP('smtp.gm
        server.ehlo()
        server.starttls()
        server.login('youremail@gmail.
        server.sendmail('youremail@gma
        server.close()

if __name__ == "__main__":
        wishMe()
        while True:
        # if 1:
            query = takeCommand().lowe

```

```
PS C:\Users\rdand\Desktop\Dollar> & C:/User
Listening...
Recognizing...
User said: open YouTube

Listening...
Recognizing...
Say that again please...
Listening...
```


File Edit Selection View Go Run Terminal Help
dollar.py - Dollar - Visual Studio Code
RUN AN... Python: Fil
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
dollar.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Google
https://www.google.com
ASUS Software Port... MyASUS Software -... McAfee LiveSafe Google Other favorites
Gmail Images
Google Search I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
India
About Advertising Business How Search works Privacy Terms Settings
Python: File (Dollar)
Ln 100, Col 42 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit

```
import pyttsx3 #pip install pyttsx
import speech_recognition as sr #p
import datetime
import wikipedia #pip install wiki
import webbrowser
import os
import smtplib

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voice
# print(voices[1].id)
engine.setProperty('voice', voices


def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

```
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
```

**Fig no 3. Images of Result**

## VI. CONCLUSION

In this Project the design and implementation of a Intelligent Personal Voice Assistance is described. The project is built using available open source software modules with visual studio code community backing which can accommodate any updates in future. The modular approach used in this project makes it more flexible and easy to integrate additional modules and features without disturbing the current system functionaries. It not only works on human commands but also it is designed for give responses to the user on the basis of query being asked or the words spoken by the user such as opening tasks and operations.

Further work will include Home appliances when they are AI-enabled, the systems will

interrelate intelligently to respond to the comfort of the occupants, including lighting, cooling, heating, and hot water systems. for example, AI technologies could cool a home by opening a window when outside temperatures are preferable and air quality is suitable, or start a ceiling fan, shade windows to reduce glare, and cycle the air conditioning. The appropriate temperature would be reached depending on who is home and in which room they are located,

This project will easily able to integrate with devices near future for a Connected Home using Internet of Things, voice command system and computer vision.

## VII. REFERENCES

1. Akash S, Neeraj Jayaram, Jesudoss A, "Desktop based Smart Voice Assistant using Python Language Integrated with Arduino", 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp.374-379, 2022.
2. Pavani Kollamudi, Bhargavi Koduru, Bharathi Poranki, Madhu Priya Golla, "Smart virtual assistant", Materials Today: Proceedings, 2021.
3. Manish Assudani, Kamran Quamar, Palash Pramod Dhongde, Pawan Sanjay Thakare, Ritik Vijaykumar Yadav, Sayyed Tarique Hasan A Rahim, "AI Based Virtual Assistance", International Journal of Advanced Research in Science, Communication and Technology, pp.381, 2022.
4. Aditya Sinha, Gargi Garg, GouravRajwani, Shimona Tayal, “Intelligent Personal Assistant”. International Journal of Informative & Futuristic Research, Volume. 4, Issue 8, April 2017.
5. Emad S. Othman .”Voice Controlled Personal Assistant Using Raspberry Pi”. International Journal of Scientific and Engineering Research Volume 8, Issue 11, November-2017.