THAKUR DEGREE COLLEGE OF SCIENCE & COMMERCE

KANDIVALI (EAST)

MUMBAI

A PROJECT REPORT ON

Speech Recognition & Personal A.I Assistant(Leo)

Designed and Developed

BY: **YOGESH SHRINATH PAL**

Submitted in partial fulfillment of

Bachelors of Science (Computer Science)

[UNIVERSITY OF MUMBAI]

Thakur Degree College of Science and Commerce
KANDIVALI (EAST)-

MUMBAI ACADEMIC YEAR 2023 - 2024

Thakur Educational Trust's (Regd.)
THAKUR COLLEGE OF SCIENCE & COMMERCE
AUTONOMOUS COLLEGE, PERMANENTLY AFFILIATED TO UNIVERSITY OF MUMBAI
NAAC Accredited Grade 'A' (3rd Cycle) & ISO 9001: 2015 (Certified)
Best College Award by University of Mumbai for the Year 2018-2019
CELEBRATING 25 YEARS OF GLORY

DATE:-

COMPUTER SCIENCE DEPARTMENT

(2023-2024)

Certificate of Approval

This is to certify that the project work entitled "Face Recognition Attendance System Software " is prepared by Ashish Pal Indrajeet a student of "Third Year Bachelor Of Science (Computer Science)" course of University of Mumbai, which is conducted by our college.

This is the original study work and important sources used have been duly acknowledged in the report. The report is submitted in partial fulfillment of B.Sc. (Computer Science) course as per rules of University of Mumbai.

MISS: SIDDHI MA`AM
Project Guide

MR. ASHISH TRIVEDI
Head of Department

External Examiner

INDEX

13	**Advantage**	
14	**Reference**	
15	**Conclusion**	
	THANK YOU	

ACKNOWLEDGEMENT

Achievement is finding out what you would be doing rather than what you have to do. It is not until you undertake such a project that you realize how much effort and hard work it really is, what are your capabilities and how well you can present yourself or other things. It gives me immense pleasure to present this report towards the fulfillment of my project.

It has been rightly said that we are built on the shoulder of others. For everything I have achieved, the credit goes to all those who had helped me to complete this project successfully.

I take this opportunity to express my profound gratitude to management of Thakur Degree College of Science & Commerce for giving me this opportunity to accomplish this project work.

I am very much thankful to **Dr.Mrs. C. T. Chakraborty** – Principal of Thakur College for their kind co-operation in the completion of my project.

A special vote of thanks to my faculty, Mr. Ashish Trivedi who is our HOD & also our project guide **Miss:Siddhi ma`am** for their most sincere, useful and encouraging contribution throughout the project span, without them we couldn't start and complete the project on time.

Finally, I would like to thank all my friends & entire Computer Science department who directly or indirectly helped me in completion of this project & to my family without whose support, motivation & encouragement this would not have been possible.

(YOGESH SHRINATH PAL)

PRELIMINARY INVESTIGATION

Organizational Overview

The Thakur College of Science and Commerce (TCSC) is a college in Kandivali in Mumbai of Maharashtra, India running by Thakur Educational Trust.

Thakur College was started in 1992 to serve the needs of students passing SSC examination from the schools around Kandivali area and Thakur Vidhya Mandir which has already established itself as one of the schools in the area. It offers courses at primarily the higher secondary and under-graduate levels. The courses at the undergraduate and postgraduate level are offered in affiliation with Mumbai University, Mumbai. An ISO 9001:2008 College with A grade as assessed by the National Assessment and Accreditation Council NAAC.

Name : Thakur College Of Science & Commerce

Founded :1997 Address : Thakur Village, Kandivali (East) Mumbai – 400101

Contacts : 022-2846 2565 / 2887 0627

Motto : Journey towards Excellence

Email : Helpdesk@tcsc.org.in

Introduction

Speech Recognition

Speech recognition is the process by which a computer (or other type of machine) identifies spoken words. Basically, it means talking to your computer and having it correctly recognize what you are saying.

The following definitions are the basics needed for understanding speech recognition technology.

- Utterance.
- Speaker Dependance.
- Vocabularies.
- Accuract.
- Training.

Personal A.I Assistant (Leo)

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.

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, pyautogui, pyQt etc. I have created a live GUI for interacting with the Leo as it gives a design and interesting look while having the conversation.

ABSTRSCTION

Speech Recognition

Speech Recognition Simply is the process of converting spoken input to text. It is also known as Speech-to-Text and Voice Recognition. Technically Speech recognition is the process of converting an acoustic signal captured by a microphone or a telephone to set of word.

The objective of voice recognition is to recognize Who is speaking. The Speech recognition aims at understanding and comprehending What was spoken. It is Used to identify a person by analyzing its tone, voice, pitch, and accent. It is used in hand-free computing, map or menu navigation.

Functionalities

- Easy Speech Recognition from Microphone.
- Translate Spoken words into text using closed captions

  To enable a person with hearing loss to understand

  What others are saying.

Personal A.I Assistant (Leo)

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 SpeechRecognition 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 favourite 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.

Functionalities

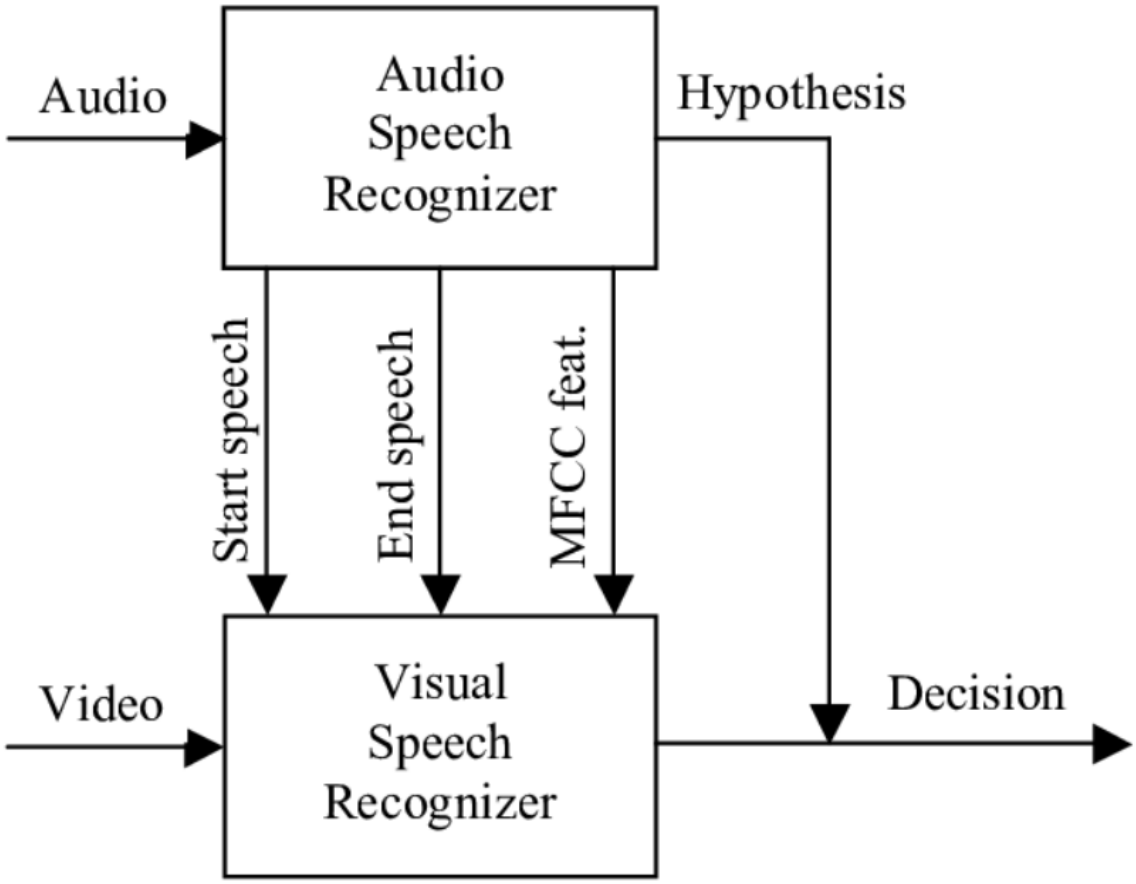
- It can open command prompt, your Favourite IDE, notepad etc.
- It can do Wikipedia searches for you.

- It can do Wikipedia searches for you.
- It can do Google searches for you.
- It can do YouTube searches for you.
- It can do Instagram Account searches for you.
- It can open websites like Google, YouTube, etc., in a web browser.
- It can give weather forecast.
- It can have some basic conversation.

Now the basic question arises in mind that how it is an AI? The virtual assistant that I have created is like if it is not an A.I, 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.I.

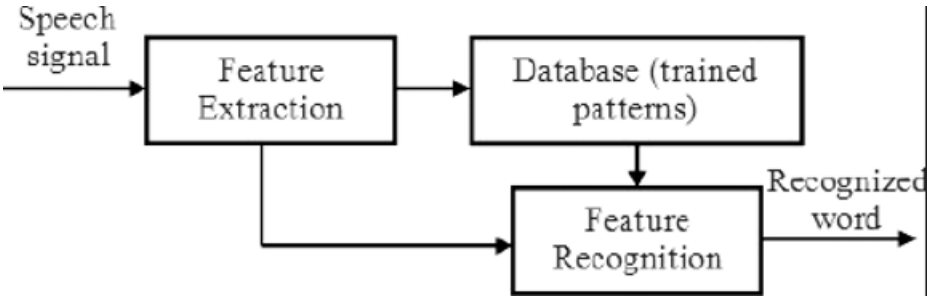
Data Flow(DAIGRAM)

The Data Flow for Speech Recognition as follow:



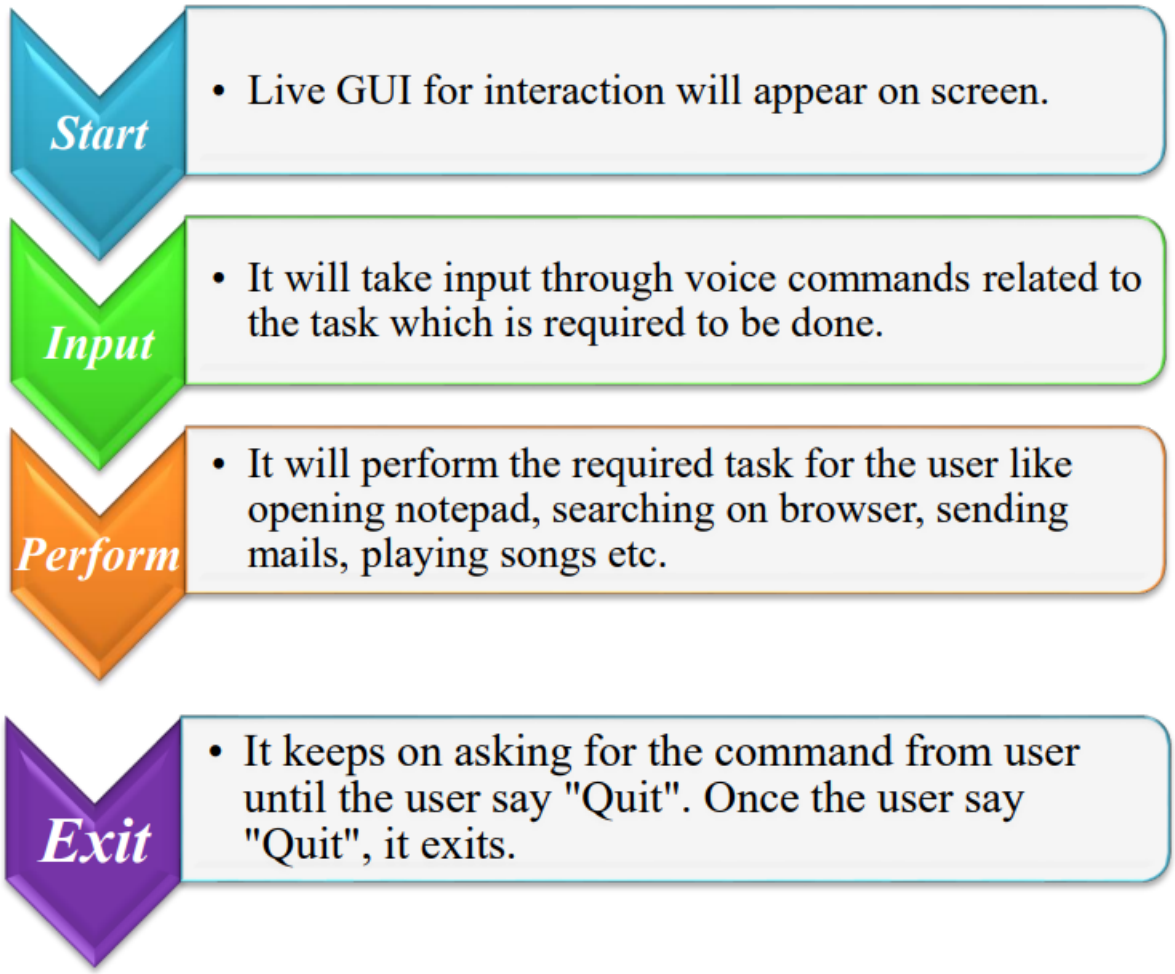
General data flow in audio-visual speech recognition system.

Speech Recognition System Block Diagram.



Speech recognition system block diagram Several different techniques for feature extraction exists, the most common being linear predictive coding (LPC) and Mel frequency cepstral coefficients (MFCC). LPC is a time- domain technique and suffers from variations in the amplitude of the speech signal due to noise [3, 4]. The preferred technique for feature extraction is MFCC [5, 6, 7] wherein the features are generated by transforming the signal into frequency domain. In general, cepstral features are more compact, discriminable, and most importantly, nearly decorrelated and therefore, they can provide higher baseline performance over filter bank features [8]. For feature recognition stage, several techniques are available including analysis methods based on Bayesian discrimination [9], Hidden Markov Models (HMM) [10], Dynamic Time Warping (DTW) based on dynamic programming [11, 12, 13], Support Vector Machines [14], Vector Quantization [15], and Neural Networks [16]. DTW is an algorithm developed by the speech recognition community to handle the matching of non-linearly expanded or contracted speech signals. In this work, we use DTW because of its simplicity in hardware implementation and it is also widely used in small-scale embedded systems (e.g. cell phones, mobile applications, etc.).

The Data Flow for A.I Assistant (Leo) as follow:



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.

Software Details

Speech Recognition

For User Interface

Html

• HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

CSS

• CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users. CSS is designed to make style sheets for the web.

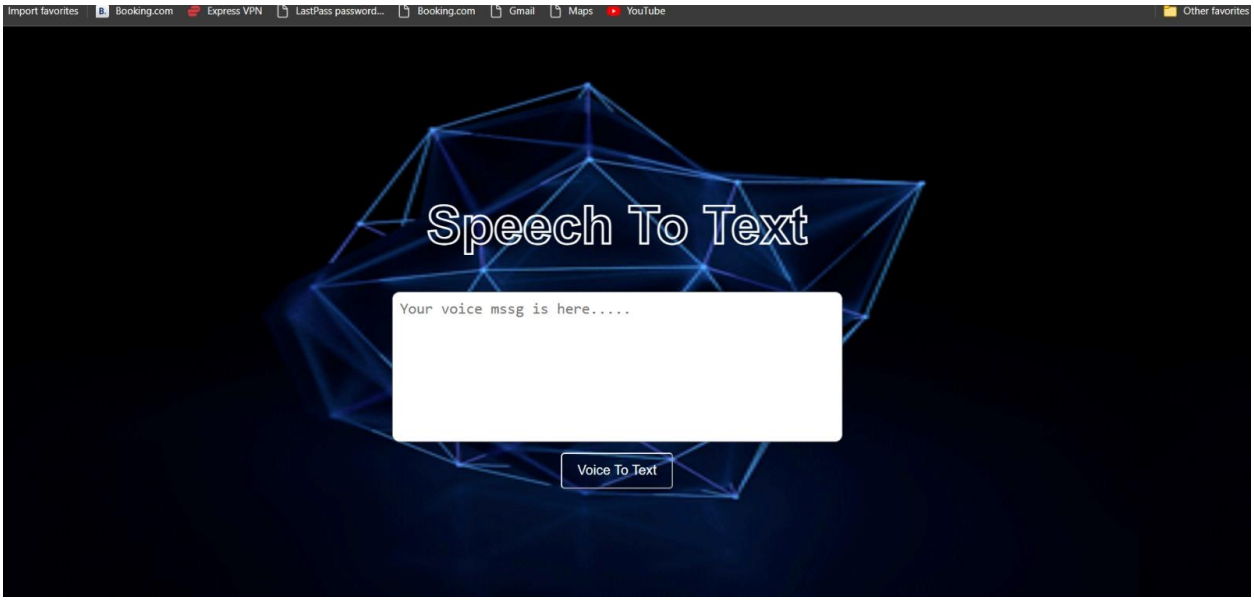
For Speech Recognition System

JavaScript

•JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus,

etc.). There are also more advanced server side versions of JavaScript such as Node.

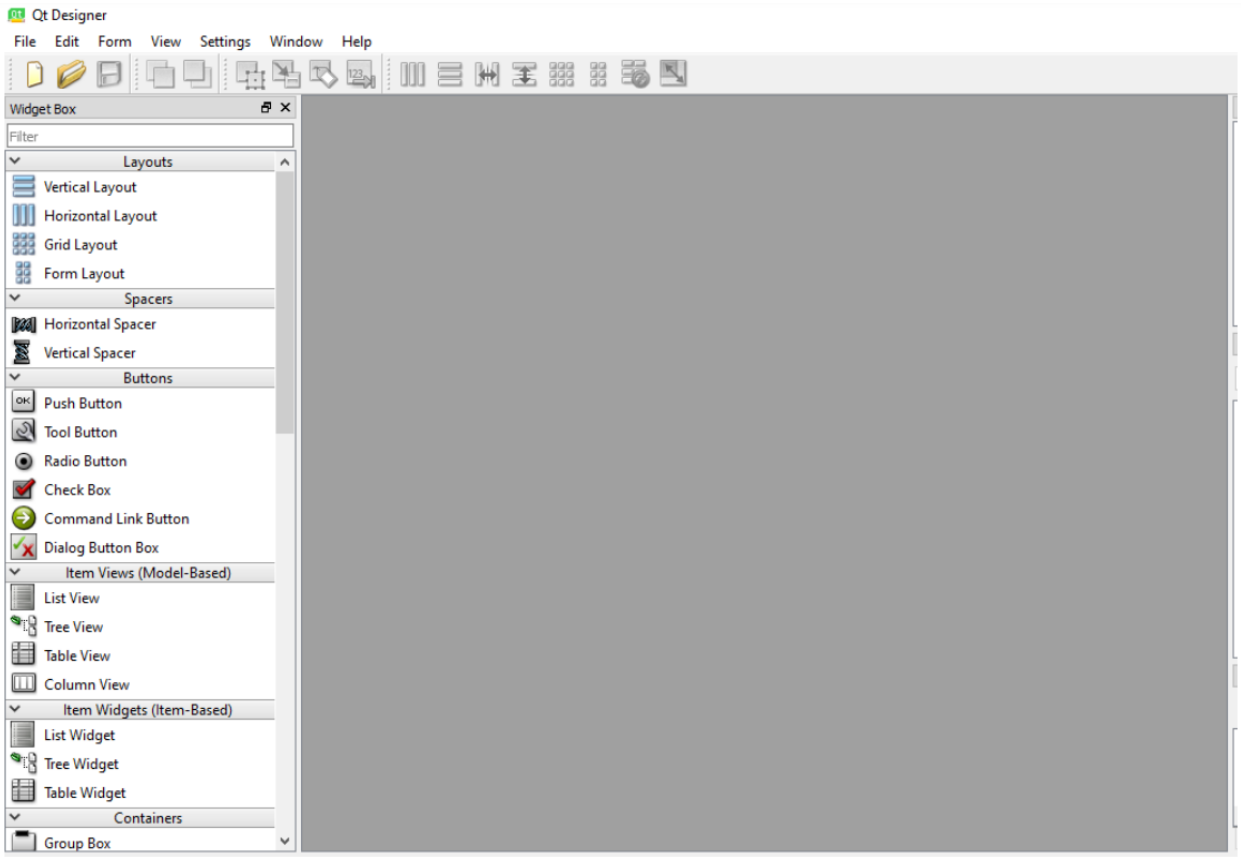
User Interface



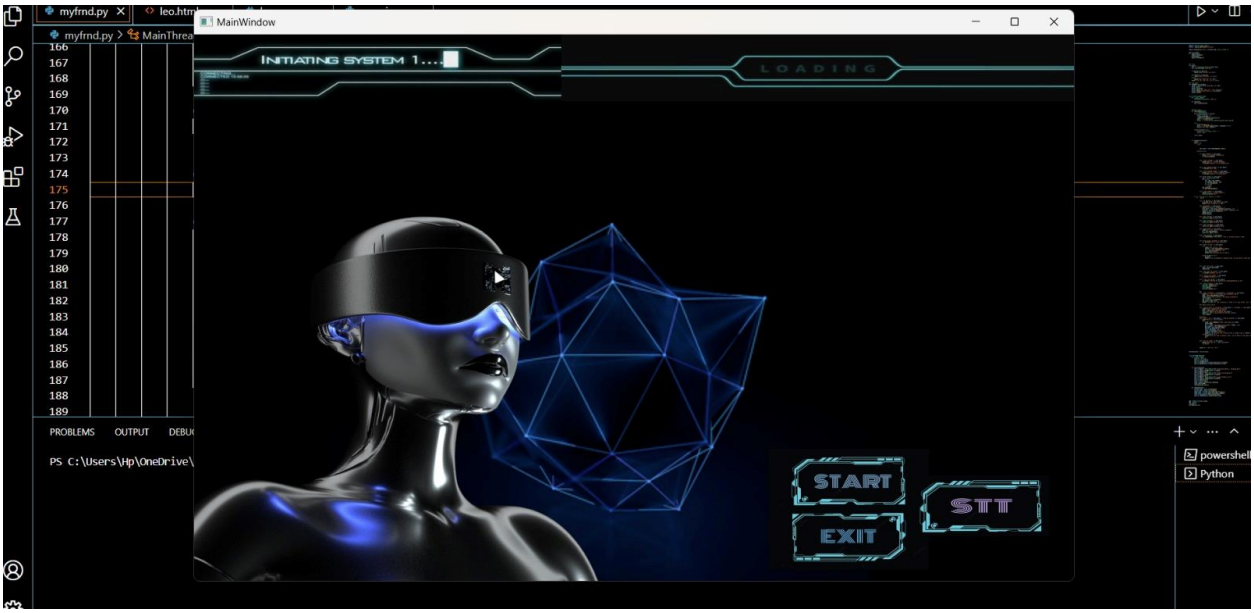
Personal A.I Assistant (Leo)

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.



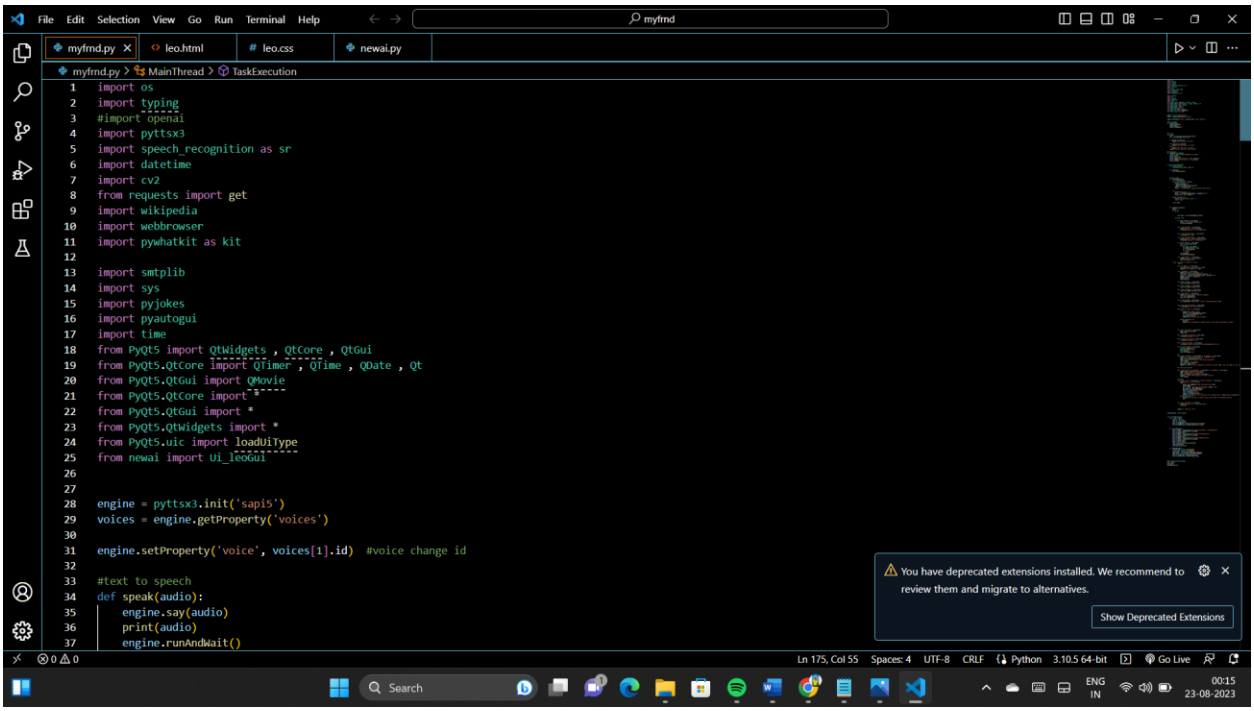
User Interface



Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.


File Edit Selection View Go Run Terminal Help
myfrnd
myfrnd.py
leo.html
leo.css
newai.py
myfrnd.py > MainThread > TaskExecution
import os
import typing
#import openai
import pyttsx3
import speech_recognition as sr
import datetime
import cv2
from requests import get
import wikipedia
import webbrowser
import pywhatkit as kit
import smtplib
import sys
import pyjokes
import pyautogui
import time
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 newai import Ui_leoGui
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id) #voice change id
#text to speech
def speak(audio):
engine.say(audio)
print(audio)
engine.runAndWait()
You have deprecated extensions installed. We recommend to review them and migrate to alternatives.
Show Deprecated Extensions
Ln 175, Col 55 Spaces: 4 UTF-8 CRLF Python 3.10.5 64-bit Go Live
Search
ENG
IN
00:15
23-08-2023

THAKUR COLLEGE OF SCIENCE AND COMMERCE

Department of Computer Science

2023-2024

Students Name: YOGESH SHRINATH PAL

Project Name: Speech Recognition & Personal A.I Assistant(leo)

College Name: Thakur College of Science and Commerce

PHASES	EXPECTED DATE OF COMPLETION	ACTUAL DATE OF COMPLETION	SIGNATURE
Preliminary Investigation			
System Analysis			
System Designing			
System Coding			
System Implementation			
Report Submission			

Python Libraries

In My personal A.I Assistant (Leo) Python Libraries were used:

- pyttsx3:

It is a python library which converts text to speech.

- SpeechRecognition:

It is a python module which converts speech to text.

- pywhatkit:

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

- Datetime:

This library provides us the actual date and time.

- Wikipedia:

It is a python module for searching anything on Wikipedia by taking user command.

- Smtplib:

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

- pyPDF2:

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

- Pyjokes:

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

- Webbrowser:

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

- Pyautogui:

It is a python libraries for graphical user interface.

- os:

It represents Operating System related functionality.

- sys:

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

Implementation & Work Details

Speech Recognition

- Pay attention to the sample rate. As we've mentioned before, audio has characteristics such as sample rate, number of channels, etc. ...
- Normalize recording volume. ...
- Improve recognition of short words. ...
- Use noise suppression methods only when needed.

Speech recognition, or speech-to-text, is the ability of a machine or program to identify words spoken aloud and convert them into readable text.

Voice recognition software is used to convert spoken language into text by using speech recognition algorithms. It can be used by people with disabilities, for in-car systems, in the military, and also by businesses for dictation, or to convert audio and video files into text.

Personal A.I Assistant (Leo)

Leo, a desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favorite IDE with the help of a single voice command. Jarvis 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.

Real Life Application

• Saves time: JARVIS 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 a set of tasks.

• 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 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.

• 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, i.e. human understandable language, English. So user finds its reaction in an informed and smart way.

• 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.

• 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.

Data Implementation & Program Execution for Leo

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:

Libraries & Packages

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

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

- pywhatkit: It is python library to send WhatsApp message at a particular time with some additional features.
- Datetime: This library provides us the actual date and time.
- Wikipedia: It is a python module for searching anything on Wikipedia.
- Smtplib: Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.
- Pyjokes: It is a python libraries which contains lots of interesting jokes in it.
- Webbrowser: It provides interface for displaying web-based

documents to users.

- Pyautogui: It is a python librariy for graphical user interface.
- os: It represents Operating System related functionality.
- sys: It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

Functions

- takeCommand(): The function is used to take the command as input through microphone of user and returns the output as string.
- wishMe(): This function greets the user according to the time like Good Morning, Good Afternoon and Jai Shree Ram.
- 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.

System testing

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

- FUNCTIONALITY

In this we check the functionality of the system whether the system performs the task which it was intended to do. To check the functionality each function was checked and run, if it is able to execute the required task correctly then the system passes in that particular functionality test. For example to check whether Leo can search on Google or not, as we can see in the figure 7.1, user said “Open Google”, then Jarvis asked, ”What should I search on Google?” then user said, “What is Python”, leo open Google and searched for the required input.

- 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 each 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.

•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.

- SECURITY

The security testing mainly focuses on vulnerabilities and risks. As leo 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.

Individual Contribution

The project titled “A.I. DESKTOP VOICE ASSISTANT: JARVIS” was designed by me individually. From installing of all the packages, importing, creating all the necessary functions, designing GUI in PyQT and connecting that live GUI with the backend, was all done by me individually.

I, myself have done all the research before making this project, designed the requirement documents for the requirements and functionalities, wrote synopsis and all the documentation, code and made the project in such a way that it is deliverable at each stage.I have created the front end (.ui file) of the project using PyQt designer, the front end comprises of a live GUI and is connected with the .py file which contains all the classes and packages of the .ui file. The live GUI consists of moving GIFs which makes the front end attractive and user friendly.

I have written the complete code in Python language and in PyCharm IDE from where it was very easy to install the packages and libraries, I have created the functions like takeCommand(), wishMe() and taskExecution() which has the following functionalities, like takeCommand() which is used to take the command as input through microphone of user and returns the output as string, wishMe() that

greets the user according to the time like Good Morning, Good Afternoon and Good Evening and taskExecution()which contains all the necessary task execution definition like many conditions in if condition like “open Google”, “open notepad”, “search on Wikipedia” ,”play music” and “open command prompt” etc.

While making this project I realized that with the advancement JARVIS 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.

At last, I have updated my report and completed it by attaching all the necessary screen captures of inputs and outputs, mentioning the limitations and scope in future of this project.

System Coding

For Speech Recognition

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-
scalable=1">
<title>STT</title>
<link rel="stylesheet" type="text/css" href="leo.css">
</head>

<body>
    <div class="hello">

        <div class="voice_to_text">
            <h1>Speech To Text</h1>
            <textarea row="5" cols"50 placeholder="Your voice mssg is here....."
id="convert_text"></textarea>

        <button onclick="voice()">Voice To Text</button>
        <button onclick="stopVoice()">Stop</button>

        </div>
    </div>







<script>
    function voice(){
        var recognition = new webkitSpeechRecognition();
        recognition.lang = "en-GB";
        recognition.onresult = function(event){
            console.log(event);
            document.getElementById("convert_text").value =
event.results[0][0].transcript;
```

```
        }
        recognition.start();
    }
    function stopVoice(){
        if (recognition){
            recognition.stop();

        }
    }



</script>
</body>
</html>

*,*:after,*:before{
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  -ms-box-sizing: border-box;
  box-sizing: border-box;
}
body{
  font-family: arial;
  font-size: 16px;
  margin: 0;
  background:linear-gradient(to right bottom, #472251, #010b28);
  color: #000;
    background: url(back.gif);
    background-size: 1550px 800px;

  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
}




.voice_to_text{
  width: 600px;
```

```
  align-items: center;
  text-align: center;
}
h1{
    color: transparent;
    font-size: 70px;
    -webkit-text-stroke: 2.5px white;
}
#convert_text{
  width: 100%;
  height: 200px;
  border-radius: 10px;
  resize: none;
  padding: 10px;
  font-size: 20px;
  margin-bottom: 10px;
}
button{
  padding: 12px 20px;
  background: transparent;
  border-color: rgba(243, 243, 247, 0.952);
  color: #fff;
  font-size: 18px;
  cursor: pointer;
  border-radius: 5px;
}
```

For Personal A.I Assistant (Leo)

```
import os
import typing
#import openai
import pyttsx3
import speech_recognition as sr
import datetime
import cv2
from requests import get
import wikipedia
import webbrowser
import pywhatkit as kit
import smtplib
import sys
import pyjokes
import pyautogui
import time
from PyQt5 import QtWidgets , QtCore , QtGui
from PyQt5.QtCore import QTimer , QTime , QDate , Qt
from PyQt5.QtGui import QMovie
from PyQt5.QtCore import QUrl
from PyQt5.QtGui import QDesktopServices
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
from newai import Ui_leoGui


engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')

engine.setProperty('voice', voices[1].id)  #voice change id

#text to speech
def speak(audio):
    engine.say(audio)
    print(audio)
    engine.runAndWait()
```

```
#for wish
def wish():
    hour = int(datetime.datetime.now().hour)
    tt = time.strftime("%I:%M %p")

    if hour>=0 and hour<=12:
        speak(f"good morning, its {tt}")

    elif hour>12 and hour<18:
        speak(f"good afternood, its {tt}")
    else:
        speak(f"jai shree ram, its {tt}")
    speak("I am leo. what can i do for you?")

#to send email
def sendEmail(to,content):
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.ehlo()
    server.starttls()
    server.login('your email id', 'your password')
    server.sendmail('your email id' ,to,content)
    server.close()


class MainThread(QThread):
    def __init__(self):
        super(MainThread,self).__init__()

    def run(self):
        self.TaskExecution()




    #voice to text
    def takecommand(self):
        r = sr.Recognizer()
        with sr.Microphone() as source:
            print("listening...")
            r.pause_threshold = 1
            r.adjust_for_ambient_noise(source)
```

```
            audio = r.listen(source)
            #audio = r.listen(source,timeout=5,phrase_time_limit=8)

        try:
            print("Recognizing...")
            query = r.recognize_google(audio, language='en-in')
            print(f"user said: {query}")

        except Exception as e:
            #speak("say that again please...")
            return "none"

        return query



    def TaskExecution(self):
        wish()
        while True:
            if 1:


                self.query = self.takecommand().lower()

            #working tasks

                if "open notepad" in self.query:
                    npath= "C:\\Windows\\notepad.exe"
                    os.startfile(npath)


                elif "close notepad" in self.query:
                    speak("okay sir, closing notepad")
                    os.system("taskkill /f /im notepad.exe")


                elif "open command prompt" in self.query:
                    os.system("start cmd")

                elif "close command prompt" in self.query:
                    speak("okay sir, closing command prompt")
                    os.system("taskkill /f /im cmd.exe")

                elif "open camera" in self.query:
                    cap = cv2.VideoCapture(0)
```

```
                    while True:
                        ret, img = cap.read(0)
                        cv2.imshow('webcam', img)
                        k = cv2.waitKey(50)
                        if k == 27:
                            break;
                    cap.release()
                    cv2.destroyAllWindows()

                elif "close camera" in self.query:
                    speak("okay sir, closing camera")
                    pyautogui.press("esc")

            # elif "play music on spotify" in query:
                #    apath=

                elif "ip address" in self.query:
                    ip = get('https://api.ipify.org').text
                    speak(f"your ip address is {ip}")

                elif "wikipedia" in self.query:
                    speak("searching on wikipedia...")
                    self.query = self.query.replace("wikipedia", "")
                    results = wikipedia.summary(self.query, sentences = 3)
                    speak("according to wikipedia")
                    speak(results)
                    print(results)

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

                elif "open facebook" in self.query:
                    webbrowser.open("facebook.com")

                elif "open wikipedia" in self.query:
                    webbrowser.open("wikipedia.com")

                elif "open google" in self.query:
                    speak("sir, what should i search on google")
                    cm = self.takecommand()
                    webbrowser.open(f"{cm}")

                elif "send message" in self.query:
                    kit.sendwhatmsg("+9619713720", "this is testing
protocal",1,10)
```

```
                elif "play song on youtube" in self.query:
                    kit.playonyt("arjit singh playlist")

                elif "email to yogi" in self.query:
                    try:
                        speak("what should i say")
                        content = self.takecommand().lower()
                        to = "yogeshpal2005@gmail.com"
                        smtplib(to,content)
                        speak("Email has been sent to yogi")

                    except Exception as e:
                        print(e)
                        speak("sorry sir,because of network issue i am not able
to sent mail to yogi")




                elif "tell me a joke" in self.query:
                    joke = pyjokes.get_joke()
                    speak(joke)

                elif "shut down the system" in self.query:
                    os.system("shutdown /s/t 5")

                elif "restart the system" in self.query:
                    os.system("shutdown/r/t 5")

                elif "sleep the system" in self.query:
                    os.system("rundll32.exe powrprof.dll,SetSuspendState 0,1,0")

                elif "switch window" in self.query:
                    pyautogui.keyDown("alt")
                    pyautogui.press("tab")
                    time.sleep(1)
                    pyautogui.keyUp("alt")

                #ss
                elif "take screenshot" in self.query or "screenshot" in
self.query:
                    speak("sir, tell me the name of the screenshot file")
```

```
                    name = self.takecommand().lower()
                    speak("please hold a second, i am taking screenshot")
                    time.sleep(2)
                    img = pyautogui.screenshot()
                    img.save(f"{name}.png")
                    speak("i am done sir, the screenshot is saved in our main
folder. now i am ready for next command")

                #to check insta profile

                elif "check profile on instagram" in self.query or "instagram" in
self.query:
                    speak("Enter instagram id correctly")
                    name = input("Enter user name here")
                    webbrowser.open(f"www.instagram.com/{name}")
                    speak("sir here is the profile of the user {name}")
                    time.sleep(2)

                #location
                elif "where i am" in self.query or "find my location" in
self.query:
                    speak("wait sir, let me check")
                    try:
                        ipAdd = get.requests('https://api.ipify.org').text
                        print(ipAdd)
                        url = 'https://get.geojs.io/v1/ip/geo/'+ipAdd+'.json'
                        geo_requests = get.requests(url)
                        geo_data = geo_requests.json()
                        city = geo_data['city']
                        state = geo_data['state']
                        country = geo_data['country']
                        speak(f"i am not sure, but i think we are in {city} city
of {state} state in{country}")
                    except Exception as e:
                        speak("sorry sir, due to network issue i am not able to
find where we are")
                        pass


                elif "you can sleep" in self.query:
                    speak("thank you sir , have a good day")
                    sys.exit()


                #speak("sir, what can i do")
```

```
startExecution = MainThread()


class Main(QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_leoGui()
        self.ui.setupUi(self)
        self.ui.pushButton.clicked.connect(self.startTask)
        self.ui.pushButton_2.clicked.connect(self.close)
        self.ui.pushButton_3.clicked.connect(self.openLink)

    def openLink(self):

        url = QUrl("https://speechto2text.netlify.app")
        QDesktopServices.openUrl(url)




    def startTask(self):
        self.ui.movie = QtGui.QMovie("New folder/Glow Matrix - Animated.gif")
        self.ui.label_2.setMovie(self.ui.movie)
        self.ui.movie.start()
        self.ui.movie = QtGui.QMovie("New folder/initiating.gif")
        self.ui.label_4.setMovie(self.ui.movie)
        self.ui.movie.start()
        self.ui.movie = QtGui.QMovie("New folder/loading_1.gif")
        self.ui.label_5.setMovie(self.ui.movie)
        self.ui.movie.start()
        timer = QTimer(self)
        timer.timeout.connect(self.showTime)
        timer.start(1000)
        startExecution.start()

    def showTime(self):
        current_time = QTime.currentTime()
        current_date = QDate.currentDate()
        label_time = current_time.toString('hh:mm:ss')
        label_date = current_date.toString(Qt.ISODate)
        self.ui.textBrowser.setText(label_date)
        self.ui.textBrowser_2.setText(label_time)
```

```
app = QApplication(sys.argv)
leo = Main()
leo.show()
exit(app.exec_())
```

Advantage

Speech Recognition

- Talking is much faster than typing.
- You can dictate a document three times faster than you can type it.
- Dictation paired with transcription software means reduced transcription costs and a much easier workflow.
- Voice recognition software can be used by any industry.

Personal A.I Assistant (Leo)

- Improved User Experience.
- Increased Productivity.
- Competitive Advantage.
- New Revenue Streams.
- 24/7 Customer Service.
- Easy Access.
- Personalized Experience.
- Opportunities for Education.

Reference

Websites

- youtube.com
- chatgpt open ai
- Google bard ai

Conclusion

Speech Recognition

If speech recognition systems someday achieve a generally acceptable level, we may develop for example a communication system where the system may first analyze the speakers' voice and its characteristics, transmit only the character string with some control symbols, and finally synthesize the speech with individual ...

Personal A.I Assistant (Leo)

Leo 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.

THANK YOU