Virtual Desktop Assistant

Dissertation submitted
in
partial fulfillment of requirement for the award of degree of

Bachelor of Engineering
in
Electronics & Telecommunication

Submitted by

ZEESHAN RAZA **AFTAB AMIN SHEIKH**

ANKUSH SHAHU **ABHIJEET THAKUR**

Guide

Prof. Ruhina Quazi

Department of Electronics & Telecommunication Engineering

Anjuman College of Engineering & Technology, Nagpur

(Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)

Session 2021-2022

Virtual Desktop Assistant

Dissertation submitted
in
partial fulfillment of requirement for the award of degree of

Bachelor of Engineering
in
Electronics & Telecommunication

Submitted by

ZEESHAN RAZA **AFTAB AMIN SHEIKH**

ANKUSH SHAHU **ABHIJEET THAKUR**

Guide

Prof. Ruhina Quazi

Department of Electronics & Telecommunication Engineering

Anjuman College of Engineering & Technology, Nagpur

(Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)

Session 2021-2022

Declaration

We hereby declare that the work which is being presented in the thesis entitled "**VIRTUAL DESKTOP ASSISTANT**" by "Zeeshan Raza, Aftab Amin Sheikh, Ankush Shahu and Abhijeet Thakur" in partial fulfillment of requirements for the award of degree of B.E. in Electronics & Telecommunication Engineering submitted in the Department of Electronics & Telecommunication Engineering at Anjuman College of Engineering & Technology, Nagpur under Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur is an authentic record of our own work carried out during a period from 2021 to 2022 under the supervision of Prof. Ruhina Quazi. The matter presented in this thesis has not been submitted in any other University / Institute for the award of B.E. Degree.

1. Zeeshan Raza -
2. Aftab Amin Sheikh -
3. Ankush Shahu -
4. Abhijeet Thakur -

NAME & SIGNATURE OF STUDENTS

Certificate

The thesis titled “**Virtual Desktop Assistant”** submitted by **Zeeshan Raza, Aftab Amin Sheikh, Ankush Shahu and Abhijeet Thakur** for the award of degree of Bachelor of Engineering in Electronics & Telecommunication Engineering, has been carried out under my supervision at the Department of Electronics & Telecommunication Engineering of Anjuman College of Engineering & Technology, Nagpur. The work is comprehensive, complete and fit for evaluation.

Prof. Ruhina Quazi
Guide, Assistant Professor
A.C.E.T, Nagpur

Prof. Ruhina Quazi
Project Co-ordinator
A.C.E.T, Nagpur

Dr. A.S. Khan
Head, Department of ETC Engineering
A.C.E.T, Nagpur

Forwarded by –

Dr. S. M. Ali
Principal,
Anjuman College of Engineering & Technology, Nagpur

External Examiner

ACKNOWLEDGEMENT

We would like to thank several people for their help and support during the making of this project. We would like to thank our guide & project co-ordinator Prof. Ruhina Quazi for her constant guidance and supervision as well as for providing necessary information regarding the project.

We would also like to thank Dr A.S. Khan, Head of the Electronics & Telecommunication Engineering Department and Dr S.M. Ali, Principal of Anjuman College of Engineering & Technology, for their support.

Lastly, we would like to thank our families and friends for their help during the time of preparation of this project. Their help really made a difference from the start till the end of this project.

Zeeshan Raza
Aftab Amin Sheikh
Ankush Shahu
Abhijeet Thakur

VIII Semester B.E. in
Electronics & Telecommunication Engineering
A.C.E.T, Nagpur.

ABSTRACT

This project thesis discusses ways in which new technology could be harnessed to create an intelligent Virtual Desktop Assistant with a focus on user-based information. It will look at examples of intelligent programs with natural language processing that are currently available, with different categories of support, and examine the potential usefulness of one specific piece of software as a Virtual Desktop Assistant. This engages the ability to communicate socially through Natural Language Processing, holding (and analyzing) information within the context of the user. It is suggested that new technologies may soon make the idea of virtual assistants a reality. Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with natural language processing algorithms in the form of a Virtual Desktop Assistant, with basic natural language processing and the ability to function without the need for other type of human input (or programming) may already be viable.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SR. NO.	ABBREVIATIONS	FULL FORM
1	AI	ARTIFICIAL INTELLIGENCE
2	NLP	NATURAL LANGUAGE PROCESSING
3	ML	MACHINE LEARNING
4	IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
5	RAD	RAPID APPLICAION DEVELOPEMENT
6	CMD	COMMAND

INDEX

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In the 21st century, human interaction is being replaced by automation very quickly. One of the main reasons for this change is performance. There's a drastic change in technology rather than advancement. In today's world, we train our machine's to do their tasks by themselves or to think like humans using technologies like Machine Learning, Neural Networks, etc. Now in the current era, we can talk to our machines with the help of virtual assistants. There are companies like Google, Apple, Microsoft, etc. with virtual assistants like Google Now, Siri, Cortana, etc. which helps their users to control their machine by just giving input in the form of voice. These types of virtual assistants are very useful for old age, blind & physically challenged people, children, etc. by making sure that the interaction with the machine is not a challenge anymore for people. Even blind people who couldn't see the machine can interact with it using their voice only. Some of the basic tasks that are supported by most of the virtual assistants are:

- Previous year question papers (RTMNU)
- Aggregate Percentage Calculator
- Search on Wikipedia
- Stream Binaural Beats
- Open Websites
- Random Password Generator etc.

The voice assistant we have developed is a desktop-based built using python modules and libraries. This assistant is just a basic version that could perform all the basic tasks which have been mentioned above but current technology is although good in it is still to be merged with Machine Learning and Internet of Things (IoT) for better enhancements. The understanding and executing commands are still to reach a new level like the virtual assistant of the iron man named Jarvis. This is although fictional yet this is what that can be achieved using virtual assistants. All you need to do is give a command to the assistant and the rest will be performed by the assistant. With the help of voice-activated virtual assistants, there will be no need to write long codes to perform a task; the system will do so for us. The machine will work in three modes- supervised, unsupervised or reinforcement learning depending upon the usage for which the assistant is developed. This is all possible with the help of machine learning.

1.1. PROBLEM DEFINITION

- In the existing assistants, the listening time of user input is very less. So, to overcome this problem we have increased the listening time of the virtual assistant (Electro).

- If somehow the assistant is unable to understand the given command or user is unable to give proper command, then the assistant will start listening for the input again, which is not available in other assistants.

- Some commands or features, such as an aggregate percentage calculator and random password generator are not available in current assistants.

1.2. OBJECTIVES

- Features supported in the current version include playing music, emails, texts, and search on Wikipedia, or opening system installed applications, opening anything on the web browser, etc.

- Some virtual assistants specialize in offering graphic design, blog writing, bookkeeping, social media, and marketing services.

- Typical tasks a virtual assistant might perform include scheduling appointments, making phone calls, making travel arrangements, and managing email accounts.

CHAPTER 2

LITERATURE SURVEY AND REVIEW

2.1 LITERATURE SURVEY OF EXISTING SYSTEM

[1] “Speech Recognition [Internet]”. Encyclopedia Britannica Online: 2016

Speech Recognition, the ability of devices to respond to spoken commands. Speech recognition enables hands-free control of various devices and equipment (a particular boon to many disabled persons), provides input to automatic translation, and creates print-ready dictation. Among the earliest applications for speech recognition were automated telephone systems and medical dictation software. It is frequently used for dictation, for querying databases, and for giving commands to computer-based systems, especially in professions that rely on specialized vocabularies. It also enables personal assistants in vehicles and smartphones, such as Apple’s Siri.

[2] “Toys That Have a Voice”. Speech Technology Magazine [Internet].2003

- *Radio Rex:* Speech recognition has a long and spotty history in the toy industry. Radio Rex was the first commercial toy to respond to voice commands. Produced in 1922 by the Elmwood Button Co., Radio Rex predates computers by more than 20 years. Rex is a brown bulldog made of celluloid and metal that responds to its name by leaping out of its house. The dog is controlled by a spring that is held in check by an electromagnet. The electromagnet is sensitive to sound patterns containing acoustic energy around 500 Hz, such as the vowel in "Rex”.
- *Julie and Jill:* In 1987, Texas Instruments created a doll capable of responding to speech recognition utterances with spoken output. Described as the "first fully interactive" toy, Julie was a standard-size plastic doll that was indistinguishable from other dolls of the period except that it had a DSP chip that enabled it to respond to and generate speech. Julie could understand eight utterances: Julie, yes, no, OK, pretend, hungry, melody and be quiet. George Doddington, who led the team that created Julie, readily admits that Julie is prone to mis-recognitions that cause her to make inappropriate verbal responses.

[3] Speech Recognition Through the Decades: How We Ended Up with Siri. [Internet]. 2011

Looking back on the development of speech recognition technology is like watching a child grow up, progressing from the baby-talk level of recognizing single syllables, to building a vocabulary of thousands of words, to answering questions with quick, witty replies, as Apple’s supersmart virtual assistant Siri does.

Listening to Siri, with its slightly snarky sense of humor, made us wonder how far speech recognition has come over the years. Here’s a look at the developments in past decades that have made it possible for people to control devices using only their voice.

2.2 SUMMARY

Speech recognition and assistants have been around for a long time, but they have not shown much growth regarding intelligence yet. Natural language processing and reacting to phrases are not new technologies themselves, but the technology around them has grown rapidly, which has allowed assistants to expand their tasks. Fifty years ago, a voice command could only make the program solve arithmetic problems, while in the modern world one voice assistant can perform thousands of tasks. Only now that smart speakers are getting popular has voice and mood recognitions become relevant. What was learnt about artificial intelligence is that its definition can vary and be misleading in some cases. Some may refer to AI when talking about simple tasks like decision-making or speech recognition that humans can do, while others refer to a machine simulating human intelligence. The definition is open because there is not one way to describe intelligence. It is a fact that the more data a virtual assistant gathers the smarter it is, but the data may end up in wrong hands. If the user is not careful enough anyone can access their personal information or even order products without them knowing. A smart speaker may also accidentally react to something thinking it is a wake word and record the conversation. Connection to the internet always adds risks and gives room to hackers. It is important to make sure that the wi-fi is properly secured. At first glance personal assistants may seem intelligent with their various skills. However, once the base is done, the actual skills are rather easy to program. Customizing a personal assistant may seem complicated, but with certain tools and basic programming knowledge it can be done in under a day.

CHAPTER 3

METHODOLOGY

3.1 PROPOSED SYSTEM

- The system to be developed here is a Virtual Desktop Assistant.
- This system is based on Functional Programming.
- The main structure is linked with small modules containing specific task which is to be performed.
- The keyword “import” is used to link all the small modules with main structure.
- This Virtual Assistant can be used to perform multiple tasks such as getting Aggregate Percentage (RTMNU), Random Password Generator, etc.
- Basically, it provides helping hand to all the students.

3.2 BLOCK DIAGRAM

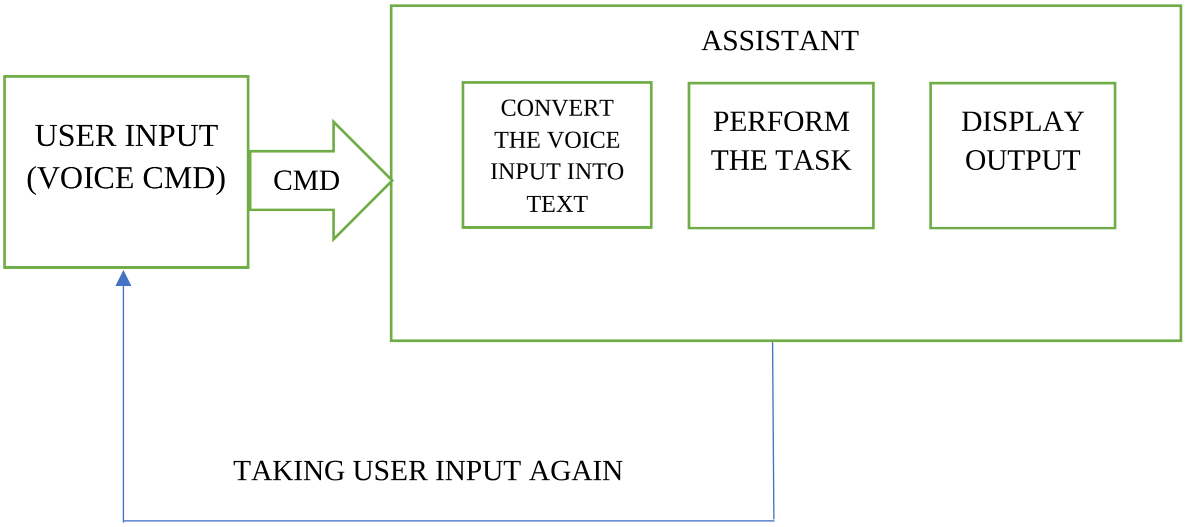


Fig. 1 Block diagram of Electro

1. **User Input (Voice Command):** Initially, the assistant electro will greet the user with respect to time and then ask the user for task and then listen to it.

2. **Assistant:** Here the assistant consists of three sub blocks i.e.
   - Convert The Voice Input into Text:
   - Perform The Task:
   - Display Output:

   a. **Convert The Voice Input into Text:**
   Here with the help of speech recognition module the electro will convert the input command into text.

   b. **Perform The Task:**
   Now according to the user input the electro will perform the task.

   c. **Display Output:**
   The result of the given task is shown here.

3. **Feedback:** The assistant will run in a loop i.e., keep asking for the input and the performing the tasks until "turn off" command is given. the turn off command will break the loop.

3.3 FLOW DIAGRAM

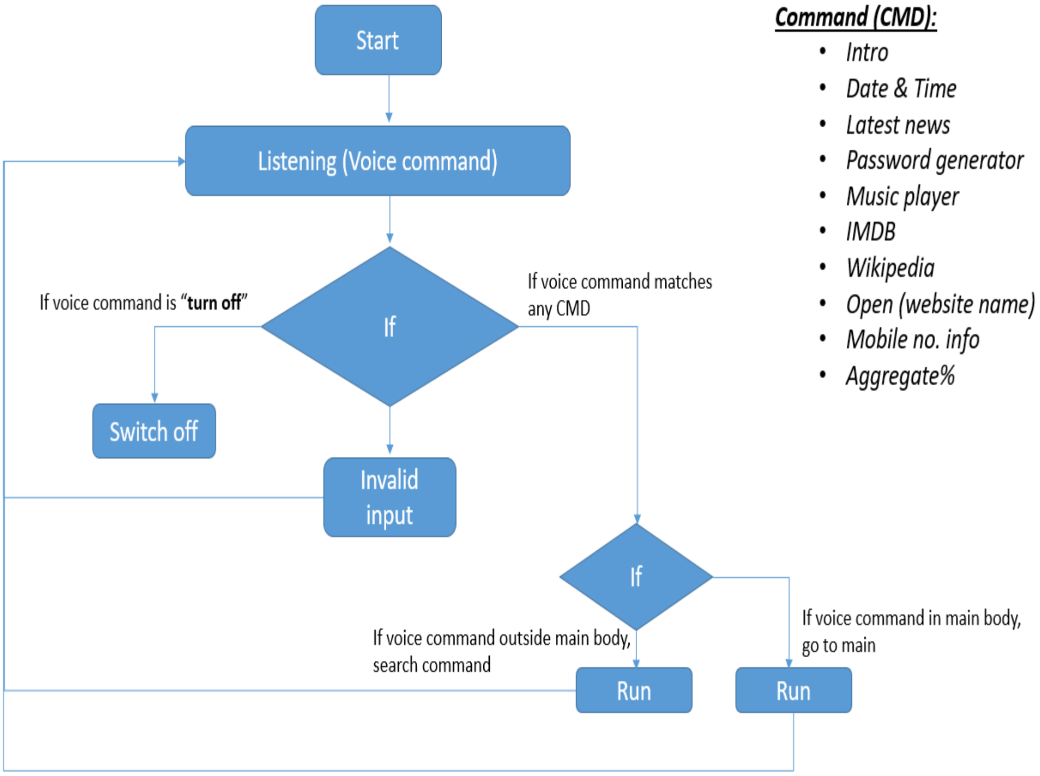

Start
Listening (Voice command)
If voice command is "turn off"
If
If voice command matches any CMD
Switch off
Invalid input
If
If voice command outside main body, search command
If voice command in main body, go to main
Run
Run
Command (CMD):
Intro
Date & Time
Latest news
Password generator
Music player
IMDB
Wikipedia
Open (website name)
Mobile no. info
Aggregate%

Figure 1 Flowchart of Virtual Desktop Assistant (Electro)

- This is the flowchart of our virtual desktop assistant.
- When we run the program, it starts with greeting the user according to time, like good morning, good afternoon.
- After this, it will start listening to the user's voice command, and recognizes the command.
- If the given command matches any of the following command then it will search for the particular function whether it is present in the main body or not.
- Wherever it is present it will execute the command and after successful execution it will go back to the listening mode.
- If it does not match or recognize the user's voice command then it says "Invalid input" and again go back to listening mode.
- To terminate the program the user will have to said turn off and it will end the loop.

CHAPTER 4

TOOLS / PLATFORM

4.1 SIMULATION ENVIRONMENT

In our project, we have used two simulation environments /IDE:

1. Visual Studio Code
2. PyCharm IDE

1. Visual Studio Code:

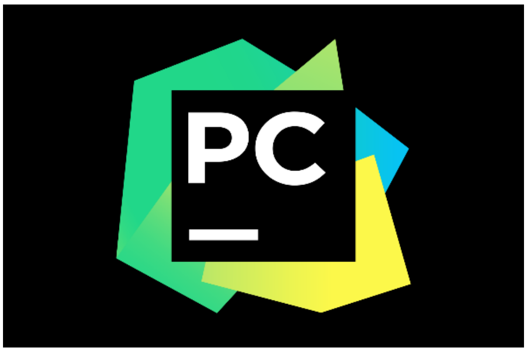
Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

2. PyCharm IDE:

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. We have used the community version of PyCharm. PyCharm supports the following versions of Python:

- Python 2: version 2.7
- Python 3: from the version 3.6 up to the version 3.11

Visual Studio Code



PyCharm IDE

4.2 SOFTWARE REQUIREMENTS

Python:

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on Rapid Application Development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), Natural language processing, data science etc. Python has a lot of libraries for every need of this project.

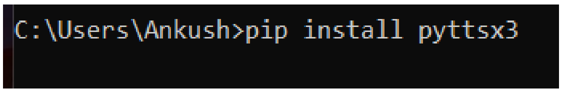
Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

Fig. Python

Modules Needed:

1. Pyttsx3:

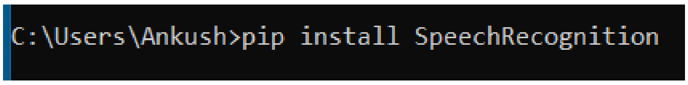
Pyttsx3 is a cross-platform text to speech library which is platform independent. The major advantage of using this library for text-to-speech conversion is that it works offline. To install this module type the below command in the terminal:



```
C:\Users\Ankush>pip install pyttsx3
```

2. Speech-Recognition:

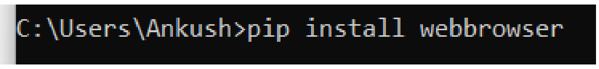
It allows us to convert audio into text for further processing. To install this module type the below command in the terminal:



```
C:\Users\Ankush>pip install SpeechRecognition
```

3. Web Browser:

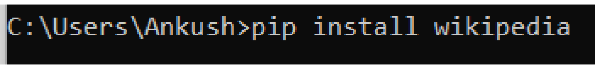
It provides a high-level interface which allows displaying Web-based documents to users. To install this module type the below command in the terminal:



```
C:\Users\Ankush>pip install webbrowser
```

4. Wikipedia:

It is used to fetch a variety of information from the Wikipedia website. To install this module type the below command in the terminal:



```
C:\Users\Ankush>pip install wikipedia
```

Created Modules:

1. **Aggregate Percentage:**

   This module is created for calculating aggregate percentage from SGPA and Total Credits of each semester.

2. **Random Password Generator:**

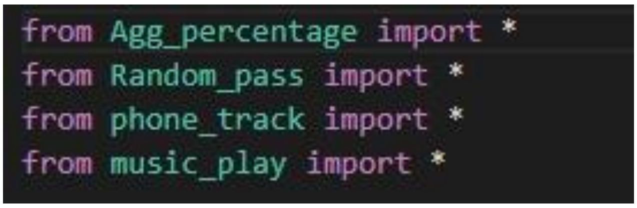
   This module is created to generate complex passwords for users to secure their different social media accounts.

3. **Phone Tracking:**

   This module is used to get user network provider and location by using mobile number.

4. **Music Player:**

   This module is used to play music and binaural beats which helps the user to increase his brain activity while studying or working.



```
from Agg_percentage import *
from Random_pass import *
from phone_track import *
from music_play import *
```

Fig. Created Modules

Other Modules:

1. Datetime
2. IMDb
3. Calendar

CHAPTER 5

DESIGN AND IMPLEMENTATION

5.1 OUTCOMES

- The mass adoption of artificial intelligence in users’ everyday lives is also fueling the shift towards voice.

- The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user’s life.

- Smart speakers are the number one way we are seeing voice being used.

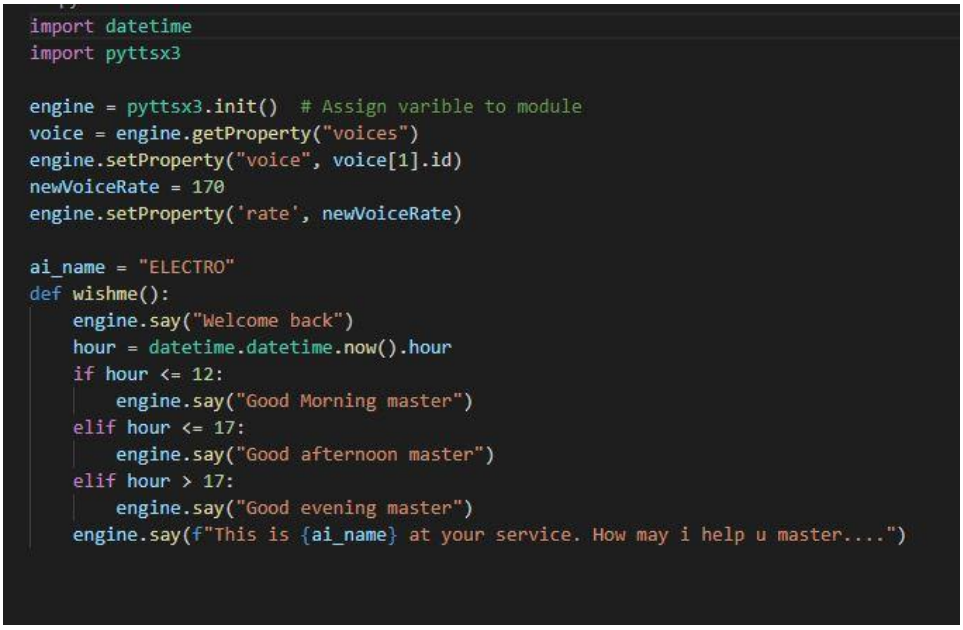
- Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years.

- The use of virtual assistants can also enhance the system of IoT (Internet of Things).

- Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually reserved for the rich and famous.

Modules:

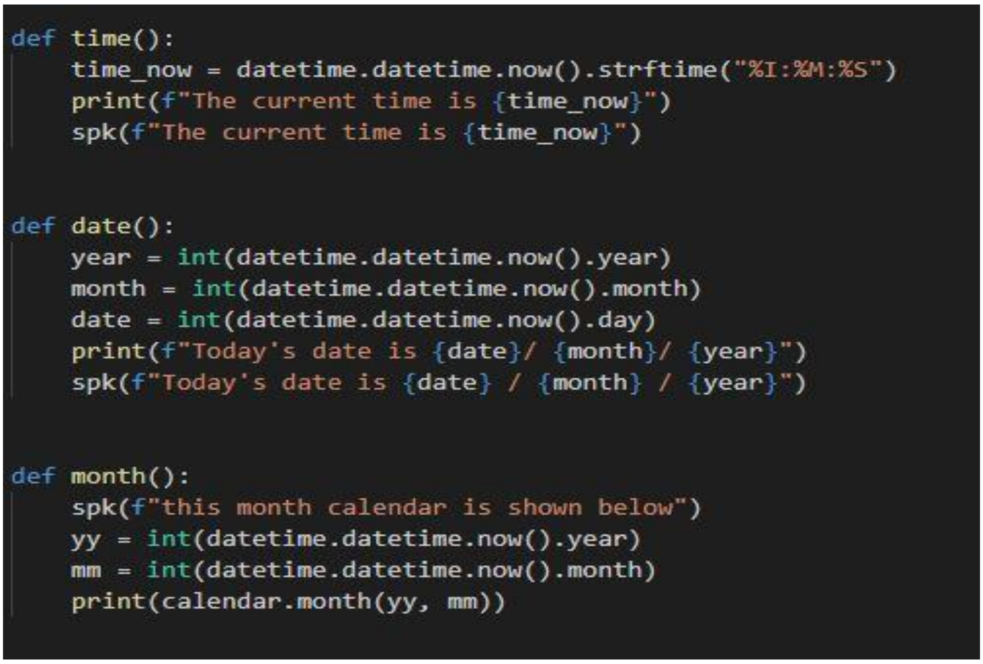
1. Greeting Module:



```python
import datetime
import pyttsx3

engine = pyttsx3.init()  # Assign varible to module
voice = engine.getProperty("voices")
engine.setProperty("voice", voice[1].id)
newVoiceRate = 170
engine.setProperty('rate', newVoiceRate)

ai_name = "ELECTRO"
def wishme():
    engine.say("Welcome back")
    hour = datetime.datetime.now().hour
    if hour <= 12:
        engine.say("Good Morning master")
    elif hour <= 17:
        engine.say("Good afternoon master")
    elif hour > 17:
        engine.say("Good evening master")
    engine.say(f"This is {ai_name} at your service. How may i help u master....")
```

Fig. Greeting Module

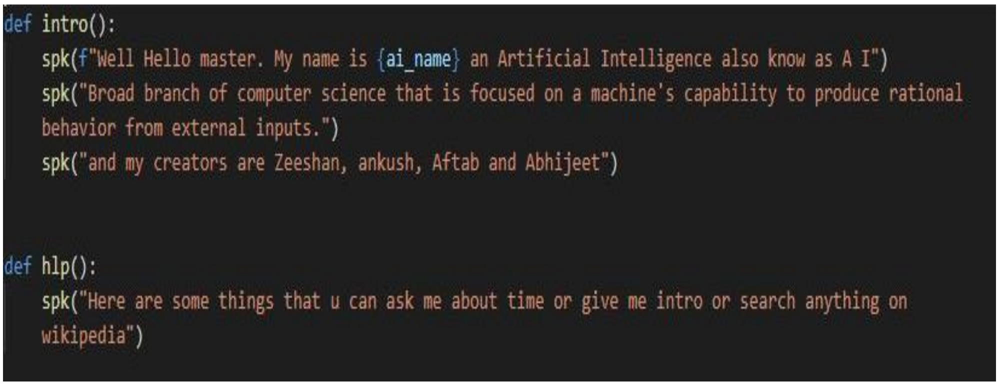
2. Datetime and Calendar modules:



```python
def time():
    time_now = datetime.datetime.now().strftime("%I:%M:%S")
    print(f"The current time is {time_now}")
    spk(f"The current time is {time_now}")


def date():
    year = int(datetime.datetime.now().year)
    month = int(datetime.datetime.now().month)
    date = int(datetime.datetime.now().day)
    print(f"Today's date is {date}/ {month}/ {year}")
    spk(f"Today's date is {date} / {month} / {year}")


def month():
    spk(f"this month calendar is shown below")
    yy = int(datetime.datetime.now().year)
    mm = int(datetime.datetime.now().month)
    print(calendar.month(yy, mm))
```

Fig. Datetime and Calendar Modules

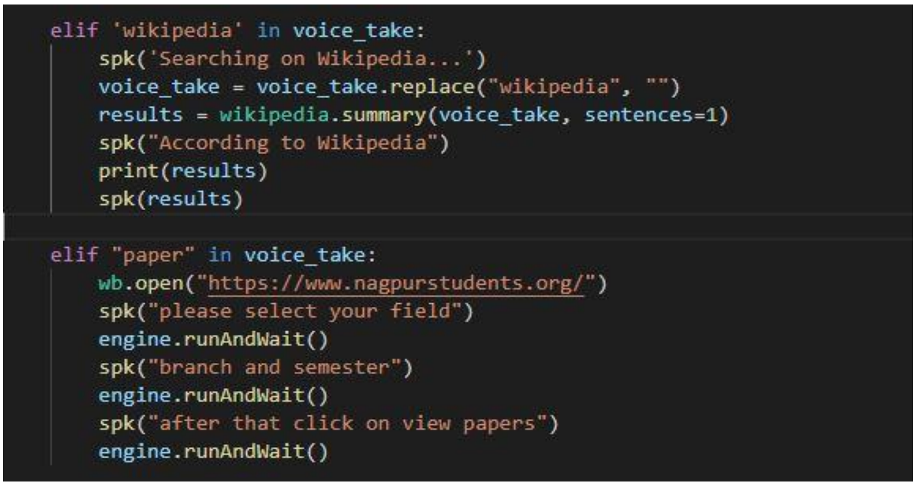
3. Introduction and Help modules:



```
def intro():
    spk(f"Well Hello master. My name is {ai_name} an Artificial Intelligence also know as A I")
    spk("Broad branch of computer science that is focused on a machine's capability to produce rational
    behavior from external inputs.")
    spk("and my creators are Zeeshan, ankush, Aftab and Abhijeet")


def hlp():
    spk("Here are some things that u can ask me about time or give me intro or search anything on
    wikipedia")
```

Fig. Introduction and Help modules

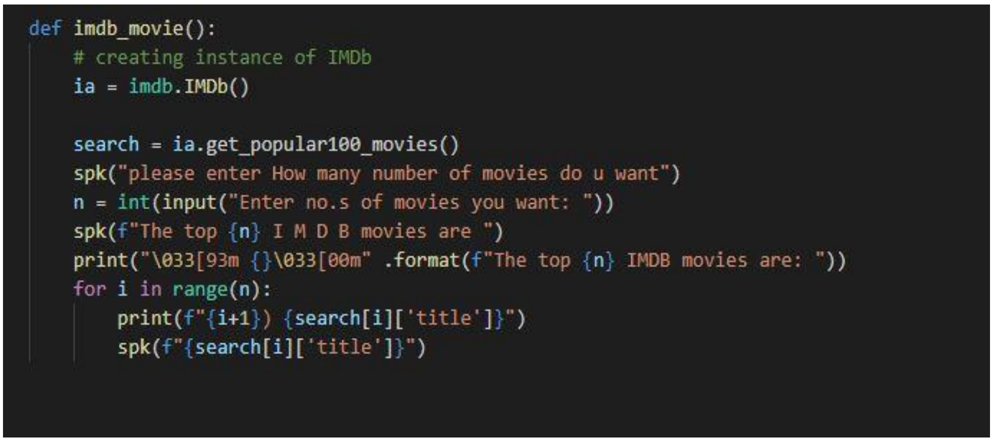
4. Wikipedia and Web browser modules:



```
    elif 'wikipedia' in voice_take:
        spk('Searching on Wikipedia...')
        voice_take = voice_take.replace("wikipedia", "")
        results = wikipedia.summary(voice_take, sentences=1)
        spk("According to Wikipedia")
        print(results)
        spk(results)

    elif "paper" in voice_take:
        wb.open("https://www.nagpurstudents.org/")
        spk("please select your field")
        engine.runAndWait()
        spk("branch and semester")
        engine.runAndWait()
        spk("after that click on view papers")
        engine.runAndWait()
```

Fig. Wikipedia and Web browser modules

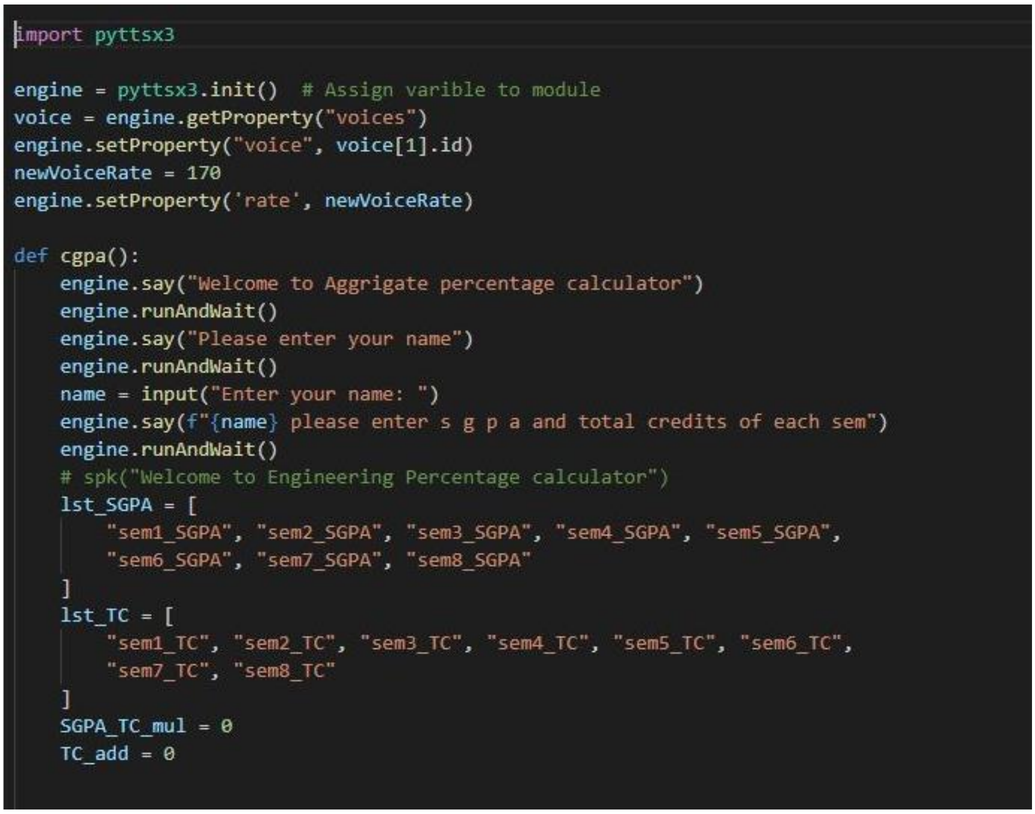
5. IMDb Module:



```
def imdb_movie():
    # creating instance of IMDb
    ia = imdb.IMDb()

    search = ia.get_popular100_movies()
    spk("please enter How many number of movies do u want")
    n = int(input("Enter no.s of movies you want: "))
    spk(f"The top {n} I M D B movies are ")
    print("\033[93m {}\033[00m" .format(f"The top {n} IMDB movies are: "))
    for i in range(n):
        print(f"{i+1}) {search[i]['title']}")
        spk(f"{search[i]['title']}")
```

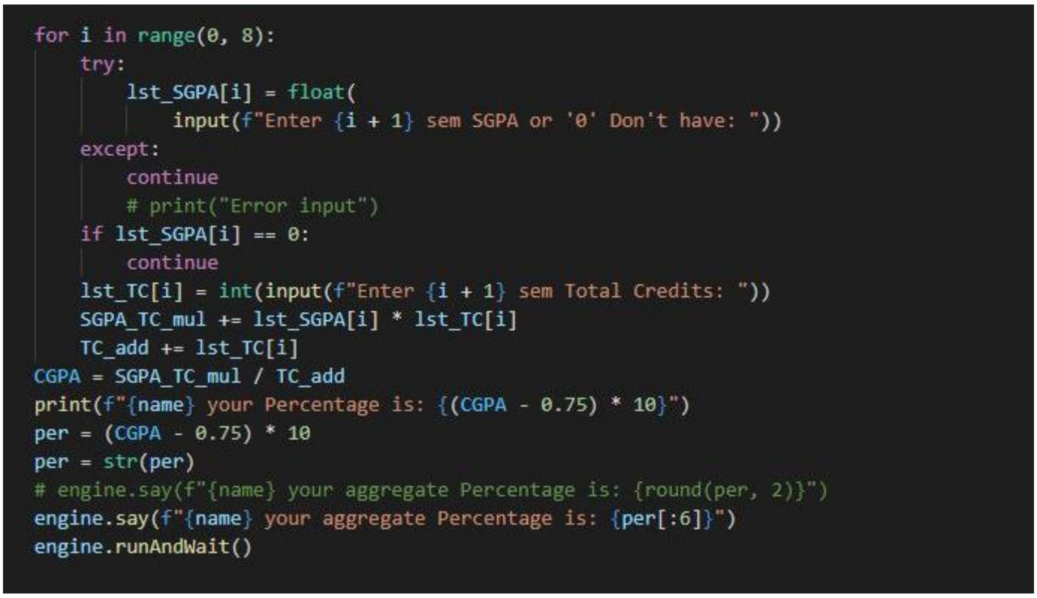
Fig. IMDb module

6. Aggregate Percentage Calculator Module:



```python
import pyttsx3

engine = pyttsx3.init()  # Assign varible to module
voice = engine.getProperty("voices")
engine.setProperty("voice", voice[1].id)
newVoiceRate = 170
engine.setProperty('rate', newVoiceRate)

def cgpa():
    engine.say("Welcome to Aggrigate percentage calculator")
    engine.runAndWait()
    engine.say("Please enter your name")
    engine.runAndWait()
    name = input("Enter your name: ")
    engine.say(f"{name} please enter s g p a and total credits of each sem")
    engine.runAndWait()
    # spk("Welcome to Engineering Percentage calculator")
    lst_SGPA = [
        "sem1_SGPA", "sem2_SGPA", "sem3_SGPA", "sem4_SGPA", "sem5_SGPA",
        "sem6_SGPA", "sem7_SGPA", "sem8_SGPA"
    ]
    lst_TC = [
        "sem1_TC", "sem2_TC", "sem3_TC", "sem4_TC", "sem5_TC", "sem6_TC",
        "sem7_TC", "sem8_TC"
    ]
    SGPA_TC_mul = 0
    TC_add = 0
```

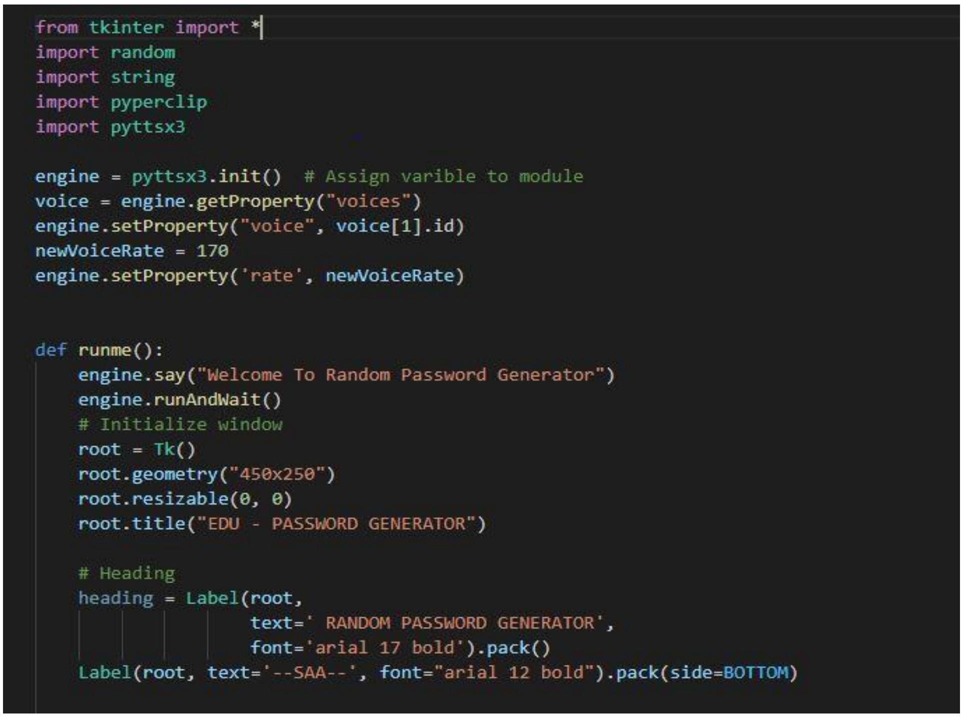
Fig. 6.1



```python
    for i in range(0, 8):
        try:
            lst_SGPA[i] = float(
                input(f"Enter {i + 1} sem SGPA or '0' Don't have: "))
        except:
            continue
            # print("Error input")
        if lst_SGPA[i] == 0:
            continue
        lst_TC[i] = int(input(f"Enter {i + 1} sem Total Credits: "))
        SGPA_TC_mul += lst_SGPA[i] * lst_TC[i]
        TC_add += lst_TC[i]
    CGPA = SGPA_TC_mul / TC_add
    print(f"{name} your Percentage is: {(CGPA - 0.75) * 10}")
    per = (CGPA - 0.75) * 10
    per = str(per)
    # engine.say(f"{name} your aggregate Percentage is: {round(per, 2)}")
    engine.say(f"{name} your aggregate Percentage is: {per[:6]}")
    engine.runAndWait()
```

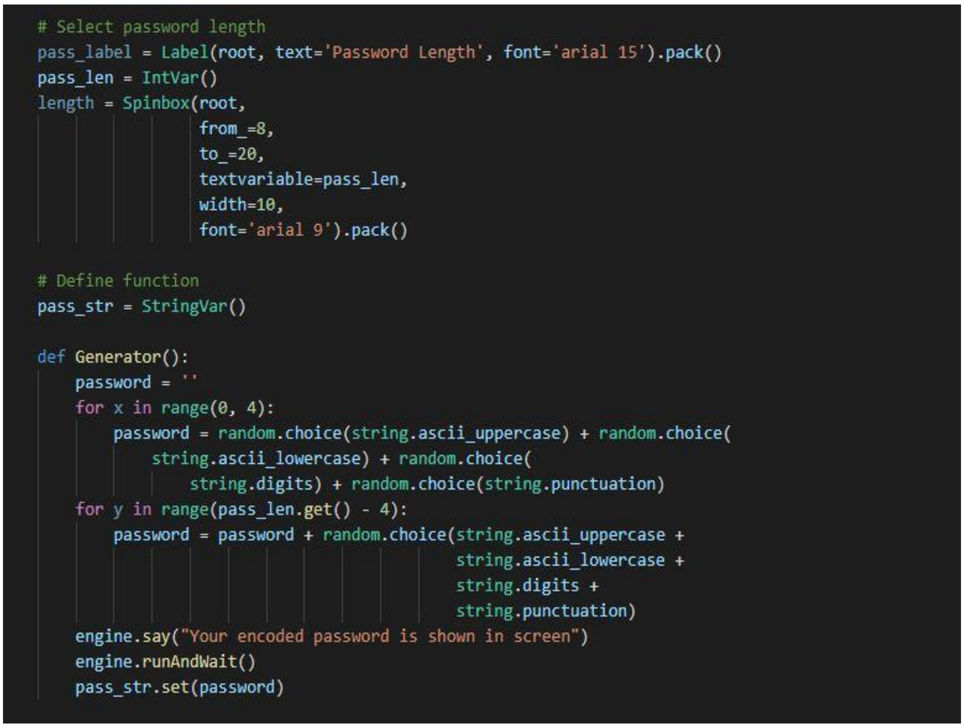
Fig. 6.2

7. Random Password Generator Module:



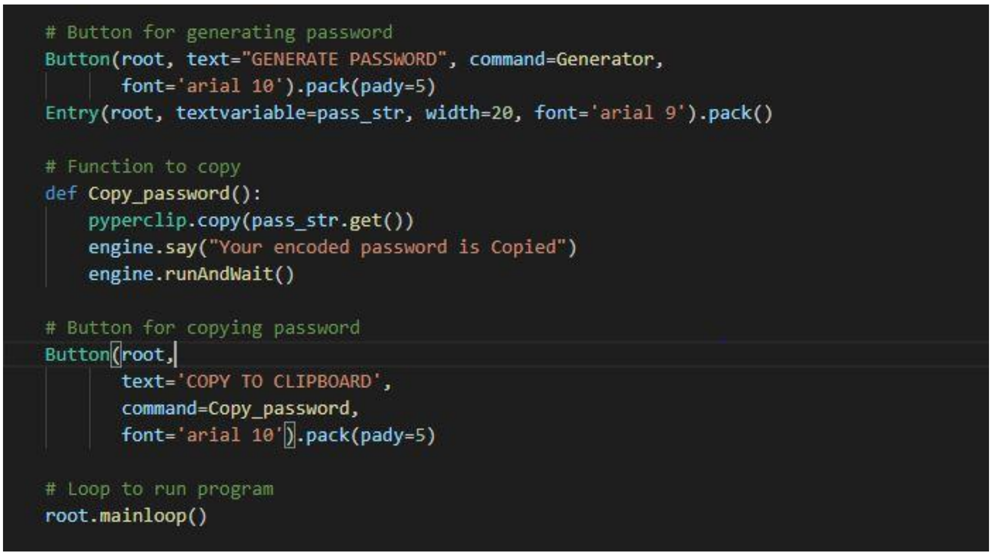
```
from tkinter import *
import random
import string
import pyperclip
import pyttsx3

engine = pyttsx3.init()  # Assign varible to module
voice = engine.getProperty("voices")
engine.setProperty("voice", voice[1].id)
newVoiceRate = 170
engine.setProperty('rate', newVoiceRate)


def runme():
    engine.say("Welcome To Random Password Generator")
    engine.runAndWait()
    # Initialize window
    root = Tk()
    root.geometry("450x250")
    root.resizable(0, 0)
    root.title("EDU - PASSWORD GENERATOR")

    # Heading
    heading = Label(root,
                    text=' RANDOM PASSWORD GENERATOR',
                    font='arial 17 bold').pack()
    Label(root, text='--SAA--', font="arial 12 bold").pack(side=BOTTOM)
```

Fig. 7.1



```
# Select password length
pass_label = Label(root, text='Password Length', font='arial 15').pack()
pass_len = IntVar()
length = Spinbox(root,
                 from_=8,
                 to_=20,
                 textvariable=pass_len,
                 width=10,
                 font='arial 9').pack()

# Define function
pass_str = StringVar()

def Generator():
    password = ''
    for x in range(0, 4):
        password = random.choice(string.ascii_uppercase) + random.choice(
            string.ascii_lowercase) + random.choice(
                string.digits) + random.choice(string.punctuation)
    for y in range(pass_len.get() - 4):
        password = password + random.choice(string.ascii_uppercase +
                                            string.ascii_lowercase +
                                            string.digits +
                                            string.punctuation)
    engine.say("Your encoded password is shown in screen")
    engine.runAndWait()
    pass_str.set(password)
```

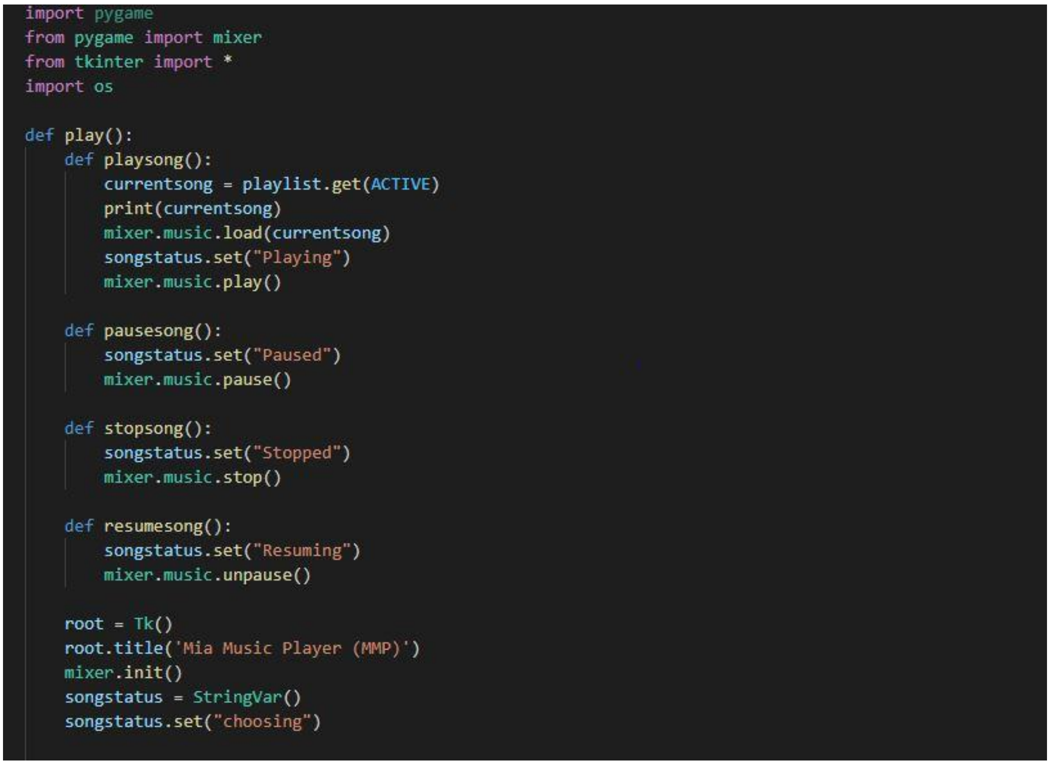
Fig. 7.2



```
# Button for generating password
Button(root, text="GENERATE PASSWORD", command=Generator,
        font='arial 10').pack(pady=5)
Entry(root, textvariable=pass_str, width=20, font='arial 9').pack()

# Function to copy
def Copy_password():
    pyperclip.copy(pass_str.get())
    engine.say("Your encoded password is Copied")
    engine.runAndWait()

# Button for copying password
Button(root,
        text='COPY TO CLIPBOARD',
        command=Copy_password,
        font='arial 10').pack(pady=5)

# Loop to run program
root.mainloop()
```

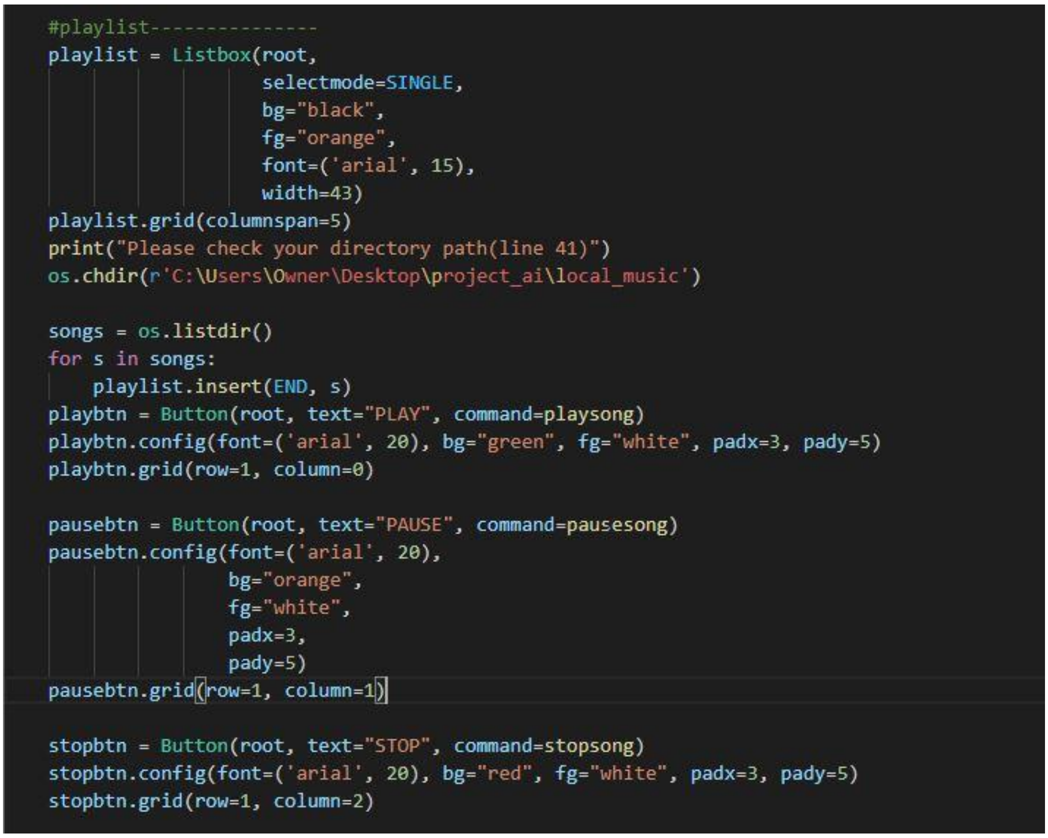
Fig. 7.3

8. Music Player Module:



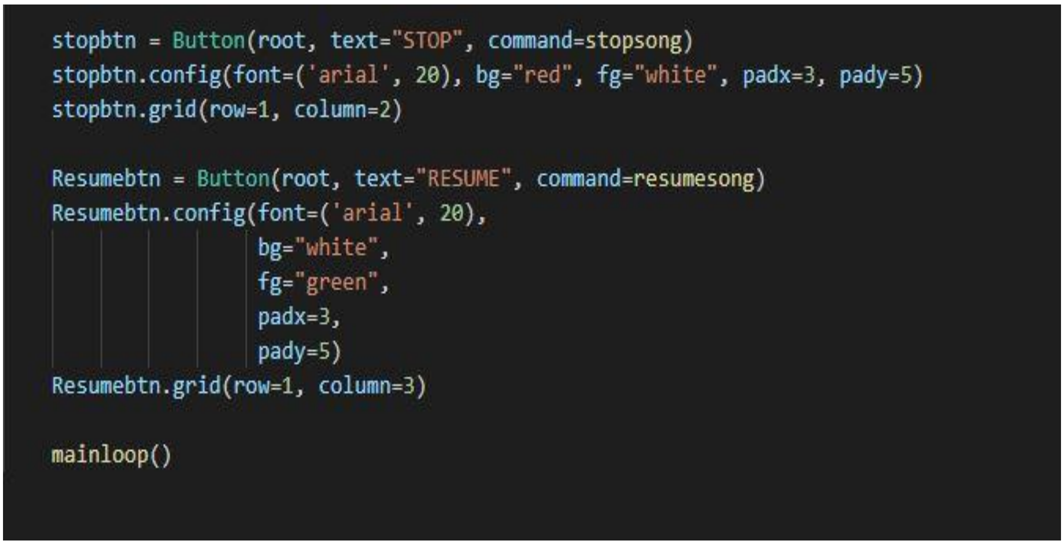
```
import pygame
from pygame import mixer
from tkinter import *
import os

def play():
    def playsong():
        currentsong = playlist.get(ACTIVE)
        print(currentsong)
        mixer.music.load(currentsong)
        songstatus.set("Playing")
        mixer.music.play()

    def pausesong():
        songstatus.set("Paused")
        mixer.music.pause()

    def stopsong():
        songstatus.set("Stopped")
        mixer.music.stop()

    def resumesong():
        songstatus.set("Resuming")
        mixer.music.unpause()

    root = Tk()
    root.title('Mia Music Player (MMP)')
    mixer.init()
    songstatus = StringVar()
    songstatus.set("choosing")
```

Fig. 8.1



```
#playlist---------------
playlist = Listbox(root,
                   selectmode=SINGLE,
                   bg="black",
                   fg="orange",
                   font=('arial', 15),
                   width=43)
playlist.grid(columnspan=5)
print("Please check your directory path(line 41)")
os.chdir(r'C:\Users\Owner\Desktop\project_ai\local_music')

songs = os.listdir()
for s in songs:
    playlist.insert(END, s)
playbtn = Button(root, text="PLAY", command=playsong)
playbtn.config(font=('arial', 20), bg="green", fg="white", padx=3, pady=5)
playbtn.grid(row=1, column=0)

pausebtn = Button(root, text="PAUSE", command=pausesong)
pausebtn.config(font=('arial', 20),
                bg="orange",
                fg="white",
                padx=3,
                pady=5)
pausebtn.grid(row=1, column=1)

stopbtn = Button(root, text="STOP", command=stopsong)
stopbtn.config(font=('arial', 20), bg="red", fg="white", padx=3, pady=5)
stopbtn.grid(row=1, column=2)
```

Fig. 8.2



```
stopbtn = Button(root, text="STOP", command=stopsong)
stopbtn.config(font=('arial', 20), bg="red", fg="white", padx=3, pady=5)
stopbtn.grid(row=1, column=2)

Resumebtn = Button(root, text="RESUME", command=resumesong)
Resumebtn.config(font=('arial', 20),
                 bg="white",
                 fg="green",
                 padx=3,
                 pady=5)
Resumebtn.grid(row=1, column=3)

mainloop()
```

Fig. 8.3

CHAPTER 6

RESULTS AND CONCLUSION

6.1 RESULT ANALYSIS

We have developed a voice assistant which can perform any kind of task in exchange of commands given by the users without any error. We have added more features like it will listen to the user’s voice only and will not be activated from environment noise. The modular nature of this project makes it easy to understand and more flexible. We can add more features without disturbing the whole program and the functionalities. The packages which are required in python programming language have been installed and the code was implemented using VS Code Integrated Development Environment (IDE).

6.2 CONCLUSION

How rapidly the time changed? If we look back around twenty year ago, Voice Recognition was in its infant stage. As when the computer system came into the existence, it was the dream to fully fledged interaction with the computer machine. Now, we can eventually talk, ask and as well order to do the assigned task.

This technology advancement is taking the world to the next level. If we think about the future competence of the voice recognition as well as the face detection, it can help the security agencies which may help to verify the details of the criminal and so forth. If we compare, two decades ago our words may very carried as we could have imagined.

Desktop Assistant has various functions as like of mobile phone like managing various applications on the voice commands. It helps to access the system hands-free and to get rid of typing chaos. By the use of facial recognition an individual can access the system, with the help of the face detection helps to secure the data, as no other person can access the system. It strategies the machine learning and help the user to access securely.

CHAPTER 7

FUTURE SCOPE

FUTURE SCOPE

What is the future vision for virtual assistants in the business world? As we see children interact effortlessly with virtual assistants, we can assume that they're here to stay and could be embedded in all aspects of our lives in the future, including our work lives. There is already some progress being made with B2B marketing virtual assistants extending the abilities of marketing professionals to improve communications. The voice technology is being tied into the Omni-channel marketing strategy that is being adopted by many businesses. For business operations, there is the potential for virtual assistants to become trusted collaborators while leveraging the predictive abilities of AI tools. Given the recent improvement of accuracy and intelligence of virtual assistants, coupled with the rising popularity of their use, it is not impossible to think that they will be transformative in business.

For virtual assistants to become fully trusted and completely incorporated into business, they must overcome a couple of hurdles. First, the security of the voice transmissions needs be addressed, so that businesses feel comfortable accessing confidential data through voice command. Second, workers’ comfort level in using virtual assistants must increase; they need to be thought of as useful tools, and not regarded as a passing gimmick. These are not impossible feats for virtual assistants, and will probably be resolved sooner than we think. In the meantime, we can enjoy the current capabilities of these digital companions – sit back, relax and ask your virtual assistant to turn on your favorite TV show and to order a pizza.

REFERENCES

REFERENCES:

1. Zwass V. Speech Recognition [Internet]. Encyclopedia Britannica Online: 2016; [cited 2019 April 7].

2. Markowitz J. Toys That Have a Voice. Speech Technology Magazine [Internet]. 2003 March [cited 2019 April 7].

3. Pinola M. Speech Recognition Through the Decades: How We Ended Up with Siri. PC World [Internet]. 2011 November 2 [Cited 2019 April 7].

4. Swapnil Saurav, Python Programing-learn and practice (2nd Edition). Ingram short title; 2018 January 1.

5. Grant Smith, Everything You Need To Ace Computer Science in One Big Fat Notebook, Workman Publishing Co., Inc, 2020 March.

APPENDICES

PROJECTEE INFORMATION

SR.NO.	PROJECTEE DETAILS	PHOTO
1.	Name: Zeeshan Raza Email: zeeshanzx26@gmail.com Mobile: 9156013781	
2.	Name: Aftab Amin Sheikh Email: aftabsheikh449@gmail.com Mobile: 9172793566	
3.	Name: Ankush Ramnaresh Shahu Email: ankushshahu1998@gmail.com Mobile: 9168397068	
4.	Name: Abhijeet Thakur Email: abhijeett64@gmail.com Mobile: 9420226819	