

A Project Report on  
**PC VOICE ASSISTANT**

Submitted in partial fulfillment of the requirements

in  
**COMPUTER DEPARTMENT**

by  
**DEEPAK YADAV (17102054)**  
**JASH VORA (17102042)**  
**RONAK JAIN (17102028)**

Under the Guidance of  
**Prof. JAYA GUPTA**



**Department of Computer Engineering**  
A.P. Shah Institute of Technology G. B. Road, Kasarvadavali, Thane(W), Mumbai-400615  
UNIVERSITY OF MUMBAI  
**Academic Year 2020-2021**

## Approval Sheet

This Project Report entitled ***“PC Voice Assistant”*** Submitted by ***“Deepak Yadav” (17102054)***, ***“Jash Vora” (17102042)***, ***“Ronak Jain” (17102028)*** is approved for the partial fulfillment of the requirement in ***Computer Department*** from ***University of Mumbai***.

(Prof. Jaya Gupta)

Guide

Prof. S.H.Malave  
Head, Computer Engineering Department

Place: A.P.Shah Institute of Technology, Thane  
Date: 18/12/2020

## CERTIFICATE

This is to certify that the project entitled “*PC Voice Assistant*” submitted by “*Deepak Yadav*” (17102054), “*Jash Vora*” (17102042), “*Ronak Jain*” (17102028) for the partial fulfillment of the requirement for award of a degree *Bachelor of Engineering* in *Computer Department*, to the University of Mumbai, is a bonafide work carried out during the academic year 2020-2021.

(Prof. Jaya Gupta)

Guide

Prof. S.H.Malave  
Head, Computer Engineering Department

Dr. Uttam D.Kolekar  
Principal

External Examiner

Place: A.P. Shah Institute of Technology, Thane  
Date: 18/12/2020

## **Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---

(Signature)

---

(Deepak Yadav,17102054)  
(Jash Vora, 17102042)  
(Ronak Jain,17102028)

Date:18/12/2020

# Contents

## **1.project conception and initiation**

- 1.1 Abstract
- 1.2 Objectives
- 1.3 Literature review
- 1.4 Problem Definition
- 1.5 Scope
- 1.6 Technology stack
- 1.7 Benefits for environment and society

## **2.Project Design**

- 2.1 proposed system
- 2.2 Design(flow of modules)
- 2.3 class diagram
- 2.4 module-1
- .....module-2.....
- .....module-n

## **3.Impelementation**

- 3.1proposed system
  - 3.1.1policies
  - 3.1.2algorithms
  - 3.1.3pseudo code
  - 3.1.4 platforms for execution

## **4.Results**

graphical and tabular representation/comparison for results

## **5.Conclusion**

conclusion and future scope

## **6.References**

## **7.Bibliography**

# **1.Project Conception- and Initiation**

## **1.1 Abstract**

The project aims to develop a personal-assistant for Computer. PC Personal Assistant draws its inspiration from virtual assistants like Google Assistant for Android, Siri for iOS. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well- defined commands. Users can interact with the assistant through voice commands. As a personal assistant, it assists the end-user with day-to-day activities like general human conversation, searching queries, reading latest news, translating words, live weather conditions, sending mail through voice. 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. PC Personal assistant is build mainly using python.

## **1.2 Objectives**

Our digital life is determined by innovations. Especially in recent years, more innovative technologies were developed to facilitate our professional and everyday life. Intelligent Personal Assistant are proved to be most important innovations in terms of easing our life and providing hands-free experience. We are building a PC Personal Assistant that works on voice commands and executes the user query.

## 1.3 Literature review

### A. Speech Synthesis

Speech synthesis is the artificial production of human speech. Speech synthesis produces audio stream as output. A speech recognizer on the other hand does opposite. It takes an audio stream as input and thus turns it into text transcription. A computer system used for speech synthesis is called a speech synthesizer, and can be implemented in software or hardware products. Synthesized speech can be created by concatenating pieces of recorder speech that are stored in a database. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

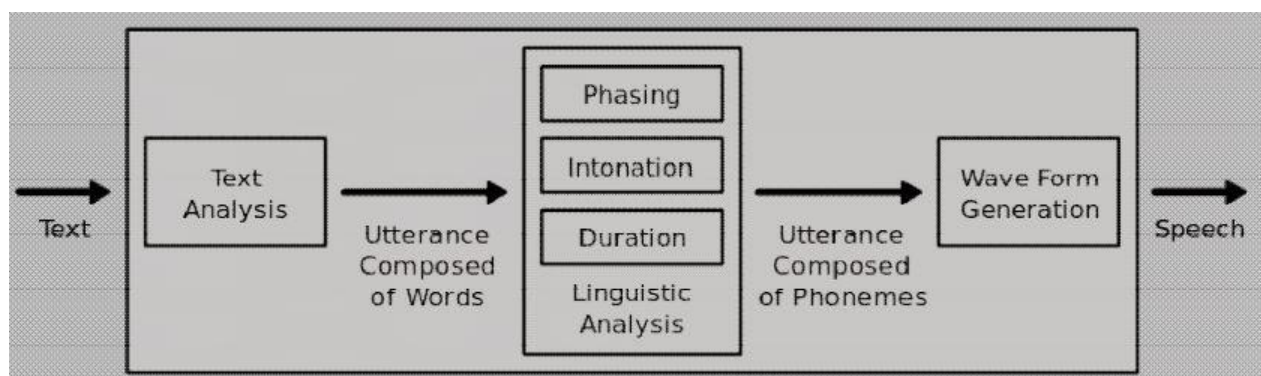


Figure 1: How a typical text-to-speech system works

The quality of speech synthesizer is judged by its similarity to human voice and by its ability to be understood clearly. The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics

## B. Synthesizer Technologies

The two primary technologies generating synthetic speech waveforms are concatenative synthesis and formant synthesis. Each technology has strengths and weaknesses, and the intended uses of a synthesis system will typically determine which approach is used. Some popular speech recognition systems are Siri, Cortana, Google Now, etc.

### 1.4 Problem Definition

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. In this virtual assistant systems if user asks web query the system redirects the user to web page and user has to manually search thereafter this can be havoc sometimes, so our Pc Voice Assistant will solve this problem by answering the web query in user interface of our app itself.

### 1.5 Scope

Presently, Jarvis is being developed as an automation tool and virtual assistant. Among the Various roles played by Jarvis are:

#### 1. Greeting

- In this features our Jarvis App will Greet you with commands like ‘Good Morning’, ‘Good Afternoon’, ‘Good Evening’ depending upon the time this function is called.
- The name of the function in our Jarvis App is “greeting\_Wish()” Whenever User says hello this will invoke greeting\_Wish() function and as per the time it will greet the user. Python datetime module is used for this Greeting functionality.
- strftime() in datetime module is used to extract the current hour and depending upon the current hour user will be greeted



## **2. Current system date and time**

- This feature helps user know its system current date and time. The name of the function in our app is “current\_Time().”
- To activate this function user has to say ‘date’ or ‘time’ and the controls goes to current\_Time() function and it will provide user with its system current date and time.

## **3. Weather related Information**

- If user wants to know weather information of particular city or his/her home city this feature comes in handy. Weather related information of any city can be provided with the help of this feature.
- We have used regular expression to extract what user said. To initiate this function user has to say current weather or weather in xyz city.
- re.group() function will extract the city name from the sentence and pass to the weather api and user can get weather related information.

## **4. Play songs from Youtube**

- We always thought that if we could listen to our favourite music by just providing its name through our voice command.
- We have provided this functionality in our project where user can listen to their favourite music just by providing name of the song by their voice command and it will be played in background.
- The name of the function in our Jarvis App is “youtube\_Music()”. The control of the program will be passed to this function when user will say ‘play me a song.’
- This function will ask user which song you want to play and by providing that user can listen to that particular song.

- We have used web parsing and regular expression to achieve this. User can also control the music through his voice. For example if user wants to increase the volume or decrease the volume or he/she wants to pause or resume the song all this they can do using voice command.

## **5. Play videos from Youtube**

- Along with youtube music if user wants to play videos of his/her choice from youtube into his/her system's VLC Media Player they can do that with the help of our "youtube\_Video()" functionality.
- Similar to youtube music user has to say 'play me a video' and youtube\_Video gets the control of program and gets executed. All the functionality like play, pause, volume increase, volume decrease are there in video play as well.

## **6. If user wants to read Reddit content**

- If user wants to search for 'Reddit' content on any topic of his/her choice he can do using his/her voice command using our project.
- User has to say search reddit and his/her topic and reddit page regarding that particular topic will be open in front of him/her.
- We have used regular expression to extract user's topic. Then that query will be added to the base url of reddit and the page will be open for user.

## **7. System Details**

- If user wants to know about his/her computer's/laptop's details he/she can easily get this by just saying 'about my system'.
- He/She can see 'System Name', 'System Architecture', 'Operating System', 'Processor Details' and some other system specific details.

## **8. Jokes**

- To lighten user's mood we have also added jokes in our app which user can listen to by saying 'tell me a joke' or 'joke'.

## **9. News For The Day**

- If user wants to keep him/her up-to-date by reading news he/she can do that with just one command i.e 'news for today' and top 5 news of the day will be available to the user.
- Python's Beautiful Library is used to convert xml data into user readable format.

## **10. Mail Transfer**

- Traditional way of writing long mails can be given some modern look by allowing user to write long mails by his/her voice commands.
- We have given this feature in our app where user can write mail to any recipient by his/her voice commands.

## **11. Shutdown / Or close the App**

- If user is done using the Jarvis app and he/she wants to quit the app he can do that by saying 'shutdown' and app will be closed.

## **12. Open any Website**

- If user wants to open any website by his/her voice command he/she can do that by just saying open 'xyz' website and our app will open that particular website for the user.

### **13. Wikipedia Search**

- Wiki Search on any topic is now very easy using our app's voice to search functionality.
- One can Wiki search on any topic of his/her choice using his/her voice command.
- Our Jarvis App will read out the content for the user.

### **14. Change Wallpaper**

- If user is bored looking at same wallpaper everyday he/she can change the wallpaper with his/her voice command.
- User has to provide with the path where the wallpapers are stored and it will be changed randomly.
- User has to just say change the wallpaper and it will be done for the user.

### **15. Translate a word**

- User can translate a word/sentence from one language to another by providing the - translate and language in which he/she wants to translate.
- User can do language translation by just saying 'translate a word' and the control of the program will be transferred to "translator\_Word()" function and word/sentence will be converted to user specific language.

### **16. Take a photo**

- User can take his/her photo by saying 'take a photo' or 'camera' and his/her camera will be open.

## **17. Empty Recycle Bin**

- User can empty his/her computer's recycle bin by saying "empty recycle bin" and it will be done for user.
- Python's 'winshell' library is used to empty recycle bin.

## **18. Take Screenshot**

- User can take screenshot of the screen by just saying 'take screenshot' or 'screenshot'.
- The Screenshot is saved in Pictures folder by default.

## **19. To-Do List**

- User can create his/her to-do list using our Jarvis Voice Assistant and add things that he/she wants the Voice Assistant to remind him/her. User can say "stop" or "done" or "that's it" to stop adding elements in the list. Once added if user want to add to existing list he/she can do that or they can also create a new list.
- Jarvis will remind user about their list when user will say 'remind the list'. If there are multiple list created by user Jarvis will ask user which list he/she wants to see.

## **20. Computational Query**

- If user want to solve or know about 'Mathematics' or 'Science and Technology' or 'Society and Culture' or 'Everyday Life' he/she can easily get that done by just saying 'who is xyz' or 'how...' or 'what...'

We have used 'Wolfram Alpha' API to extract the content that wolframalpha has.

## **1.6 Technology stack**

Front End Framework (User Interface)

- Kivy
- Kivymd

Main Framework

- Python

## **1.7 Benefits for environment and society**

1. User have a wealth of knowledge at their fingertips and can automate many time consuming activities with just a sentence or two.
2. Voice assistants can break down barriers for people with disabilities, whether sensory, physical or cognitive. Voice interaction is especially useful for those with visual impairments.
3. Virtual assistants can deliver a conversational experience that includes tone of voice – a main ingredient in building effective communication and emotional connection.
4. People genuinely enjoy speaking to home assistants, showing that a human-to-machine bond that can be created through voice.

## **2.Project Design**

### **2.1 Proposed system**

Based on the study of this system, the proposed system aims to simplify basic operations for the user, users with faulty hardware, users who could be too busy to perform certain operations themselves, elderly people and even users with sight or motor disabilities.

For example, a teacher could be scoring exam papers and remembers that he must book a flight, rather than leaving the work he's doing he could simply tell the Voice Assistant application to assist him “find nearby airports” and therefore the application will help the user open his/her browser and find airports on the brink of him and other flight details, inherently nullifying the necessity for him to try to do these himself. the appliance also will allow him (in best case scenarios) perform this task significantly faster than he would have done otherwise. the appliance also possesses speech synthesizing capabilities to offer the user the impression that he's talking and dealing with an actual assistant.

For example, a teacher could be scoring exam papers and remembers that he must book a flight, rather than leaving the work he's doing he could simply tell the Voice Assistant application to assist him “find nearby airports” and therefore the application will help the user open his/her browser and find airports on the brink of him and other flight details, inherently nullifying the necessity for him to try to do these himself. the appliance also will allow him (in best case scenarios) perform this task significantly faster than he would have done otherwise. the appliance also possesses speech synthesizing capabilities to offer the user the impression that he's talking and dealing with an actual assistant.

## 2.2 Design (Flow of Modules)

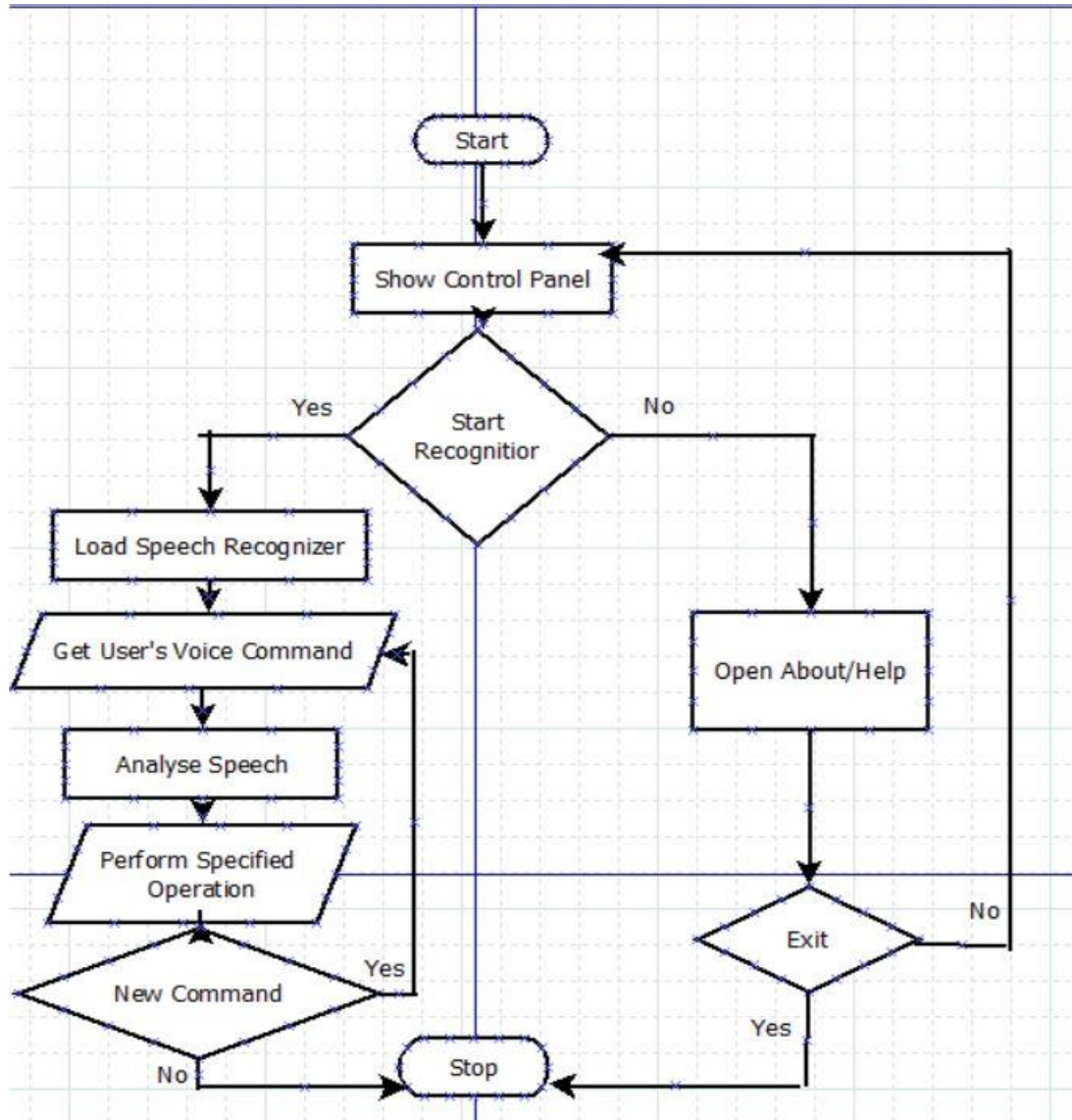


Figure 2: Flow Chart of our Project



## 2.3 Class Diagram

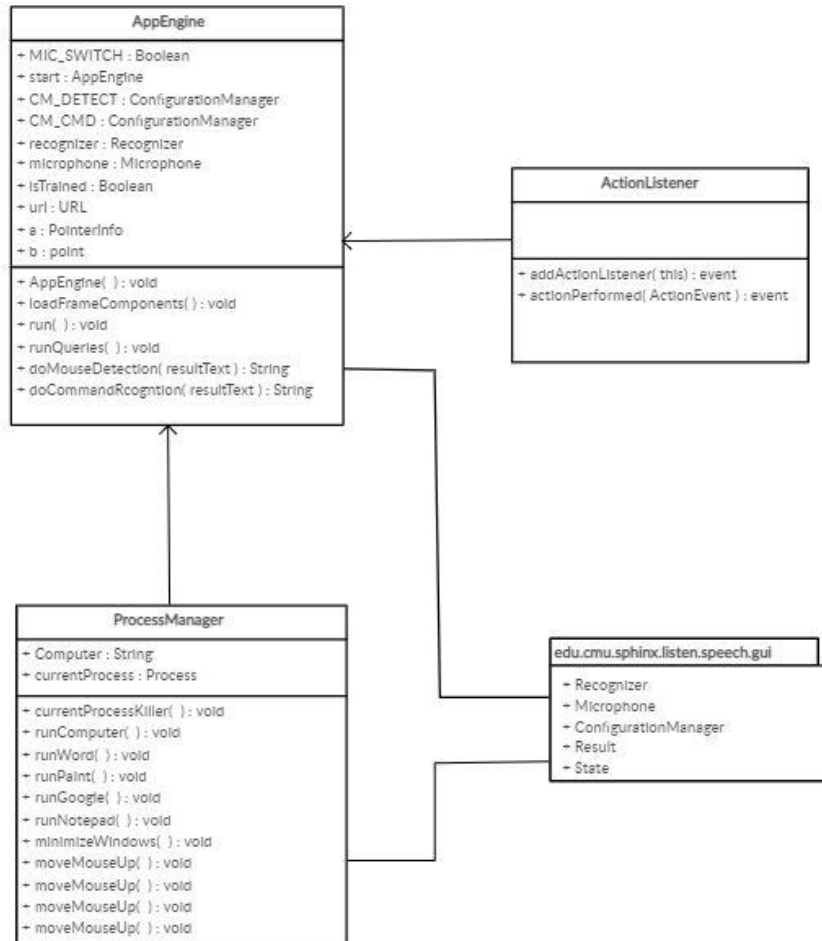


Figure 3: Class Diagram

## 2.4 Modules

### Module -1 Speech Recognition

Speech recognition, or speech-to-text, is the ability for a machine or program to identify words spoken aloud and convert them into readable text. Since we're building an Application of voice assistant, one of the most important things in this is that your assistant recognizes your voice so for that Speech Recognition module is important.

### Module -2 Wikipedia Module

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. As we all know Wikipedia is a great source of knowledge just like GeeksforGeeks we have used Wikipedia module to get information from Wikipedia or to perform Wikipedia search.

### Module -3 Urllib.parse module

Urllib.parse defines a standard interface to break Uniform Resource Locator (URL) strings up in components (addressing scheme, network location, path, parameters, query and fragment), to combine the components back into a URL. We have used urllib.parse to retrieve information from Wolfram alpha API and to get user query based youtube url.

1. `urllib.parse.urlparse()` - function is used to split url string into its components, or on combining URL components into URL string.

2. `urllib.parse.parse_qs()` - function is used to parse a query string given as a string argument. Data are returned as a dictionary. 3. `urllib.parse.urlencode()` - this function converts a mapping object, which may contain str objects, to a percent-encoded ASCII text string.

## Module - 4 RE (Regular Expression) module

A regular expression may be a sequence of characters that helps us in searching a pattern in string. Each character during a regular expression is either a metacharacter, having a special meaning, or a daily character that features a literal meaning [3]. For instance, within the regular expression b., 'b' may be a literal character that matches just 'b', while '.' may be a metacharacter that matches every character except a newline. Together, metacharacters and literal characters are often used to identify text of a given pattern or process variety of instance of it. The list of metacharacter include: . ^ \$ \* + ? { } [ ] \ | (). A regex processor is one that translates a daily expression into an indoor representation which will be executed and match against a string that represents the text that is being searched in [4]. we've used Regular expression in our project to extract youtube links.

## 3.Impelementation

### 3.1 Proposed system

Based on the study of this system, the proposed system aims to simplify basic operations for the user, users with faulty hardware, users who could be too busy to perform certain operations themselves, elderly people and even users with sight or motor disabilities.

For example, a teacher could be scoring exam papers and remembers that he must book a flight, rather than leaving the work he's doing he could simply tell the Voice Assistant application to assist him “find nearby airports” and therefore the application will help the user open his/her browser and find airports on the brink of him and other flight details, inherently nullifying the necessity for him to try to to of these himself. the appliance also will allow him (in best case scenarios) perform this task significantly faster than he would have done otherwise. the appliance also possesses speech synthesizing capabilities to offer the user the impression that he's talking and dealing with an actual assistant.

For example, a teacher could be scoring exam papers and remembers that he must book a flight, rather than leaving the work he's doing he could simply tell the Voice

Assistant application to assist him “find nearby airports” and therefore the application will help the user open his/her browser and find airports on the brink of him and other flight details, inherently nullifying the necessity for him to try to do these himself. the appliance also will allow him (in best case scenarios) perform this task significantly faster than he would have done otherwise. the appliance also possesses speech synthesizing capabilities to offer the user the impression that he's talking and dealing with an actual assistant.

### **3.1.1 Algorithms**

#### **1. Speech Recognition Algorithm:**

A speech recognition algorithm or voice recognition algorithm is used in speech recognition technology to convert voice to text. Analog to Digital Converter (Working): A microphone usually serves as an analog to digital converter. The conversion can be visualized in a graph known as a spectrogram. To create a spectrogram, three main steps are followed:

1. The sound wave is captured and placed in a graph showing its amplitude over time. Amplitude units are always expressed in decibels (dB).
2. The wave is then chopped into blocks of approximately one second, where the height of a block determines its state. Each state is then allocated a number hence successfully converting the sound from analog to digital.
3. Even when the data is digitized, something is still missing. In the speech recognition process, we need three elements of sound. Its frequency, intensity, and time it took to make it. Therefore, a complex speech recognition algorithm known as the Fast Fourier Transform is used to convert the graph into a spectrogram.

## 2. Text-To-Speech Algorithm:

A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

The quality of speech synthesizer is judged by its similarity to human voice and by its ability to be understood clearly. The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

### 3.1.2 Pseudo-Code

```
voice.jarvisResponse('What shall I play Sir?')
mysong = voice.myCommand()
if mysong:
    flag = 0
    query_string = urllib.parse.urlencode({"search_query": mysong})
    html_content = urlopen("http://www.youtube.com/results?" + query_string)
    search_results = re.findall(r"watch\?v=(\S{11})", html_content.read().decode())
    print("https://www.youtube.com/watch?v=" + search_results[0])
    url = 'http://www.youtube.com/watch?v=' + search_results[0]
    if url:
        flag = 1
    video = pafy.new(url)
    best = video.getbest()
    media = vlc.MediaPlayer(best.url)
    media.play()
    if flag == 0:
        voice.jarvisResponse('I have not found anything in Youtube ')
    while True:
        p = voice.myCommand()
        if 'volume' in p:
            self.music_Volume_Control(p, media)
            print(media.audio_get_volume())
            # v = int(input("Set :"))
            # media.audio_set_volume(v)
        elif p == 'pause':
            media.pause()
        elif p == 'resume':
            media.play()
        elif p == 'stop':
            media.stop()
            break
```

Figure 4: Youtube Video

```

voice.jarvisResponse("From which language you want to translate ")
chat_text.append('\n From which language you want to translate')
lang_in = str(voice.myCommand())
#chat_text.append(lang_in)
voice.jarvisResponse("Which word you want to translate")
chat_text.append('\n Which word you want to translate')
lang_from = str(voice.myCommand())
#chat_text.append(lang_from)
lang_supported = googletrans.LANGUAGES
voice.jarvisResponse("In Which language you want to translate")
chat_text.append('\n In Which language you want to translate')
lang_to = str(voice.myCommand())
#chat_text.append(lang_to)
for key, value in lang_supported.items():
    if lang_in == value:
        in_lang = key
for key, value in lang_supported.items():
    if lang_to == value:
        to_lang = key
translator = Translator()
text_to_translate = translator.translate(lang_from, src=str(in_lang), dest=str(to_lang))
text = text_to_translate.text
if to_lang == 'en':
    print(text)
    speech = gTTS(text=text, lang=str(to_lang), slow=False)
    speech.save("Trans.mp3")

    # convert mp3 file to wav
    sound = AudioSegment.from_mp3("Trans.mp3")
    sound.export("Trans.wav", format="wav")
    media = vlc.MediaPlayer("Trans.wav")
    media.play()

else:
    speech = gTTS(text=text, lang=str(to_lang), slow=False)
    speech.save("Trans.mp3")

```

Figure 5: Translator

```

wolfram_list = []
appid = 'WH46T8-VE953AWEYW'

base_url = 'http://api.wolframalpha.com/v2/query?'
url_params = {'input': command, 'appid': appid}

url_parse = urllib.parse.urlparse(base_url)
query = url_parse.query
data = urllib.parse.urlencode(query)
url_dict = dict(urllib.parse.parse_qs(query))
url_dict.update(url_params)
url_new_query = urllib.parse.urlencode(url_dict)
url_parse = url_parse._replace(query=url_new_query)
new_url = urllib.parse.urlunparse(url_parse)
print(new_url)
http = urllib3.PoolManager()
r = http.request('GET', new_url)
xml = r.data
# print(xml)

data_dics = {}
tree = ET.fromstring(xml)
# retrieving every tag with label 'plaintext'
for e in tree.findall('pod'):
    for item in [s for s in list(e) if s.tag == 'subpod']:
        for it in [i for i in list(item) if i.tag == 'plaintext']:
            data_dics[e.get('title')] = it.text

# TO remove None value
empty_keys = [k for k, v in data_dics.items() if not v]
for k in empty_keys:
    del data_dics[k]
# print(data_dics)

```

Figure 6: Wolfram Alpha



### 3.1.3 Platform for Execution

- Pycharm
- Anaconda

## 4.Results

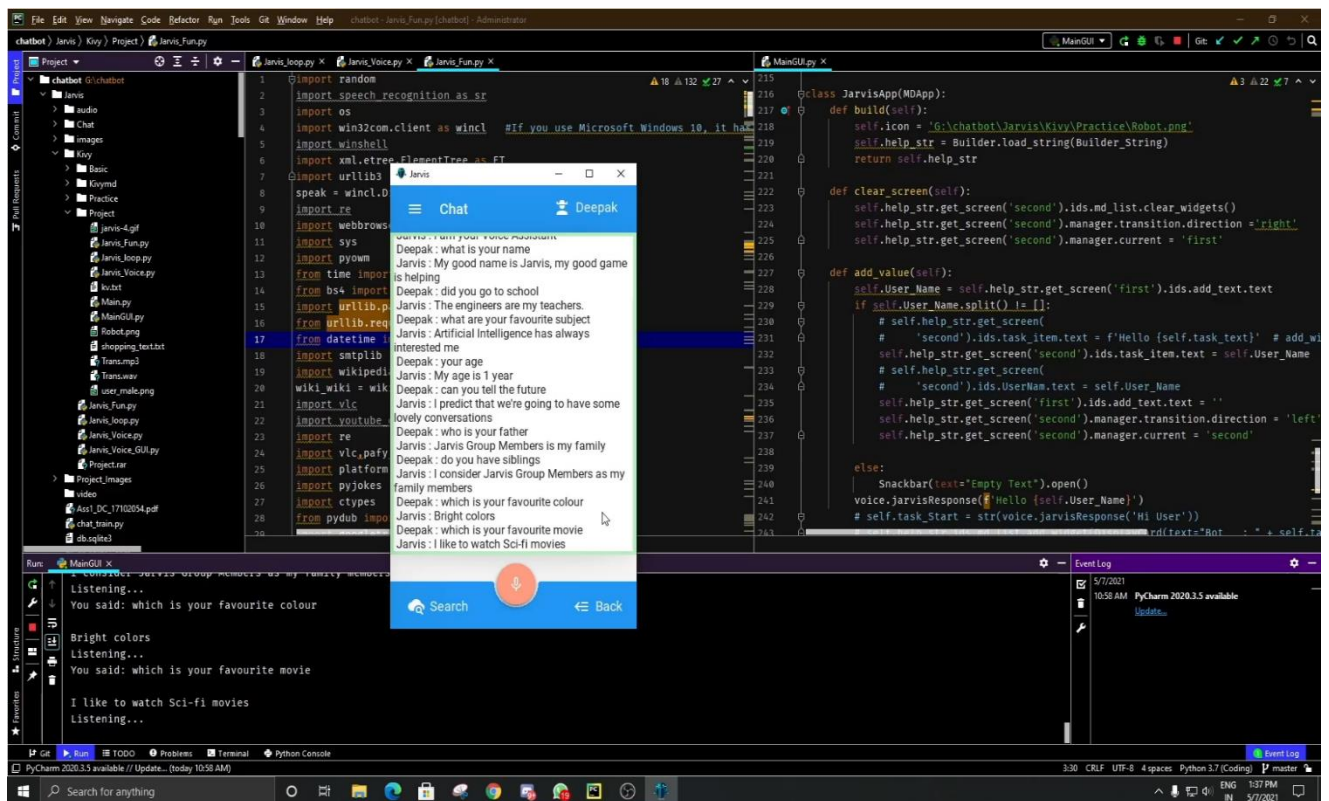


Figure 7: Result 1

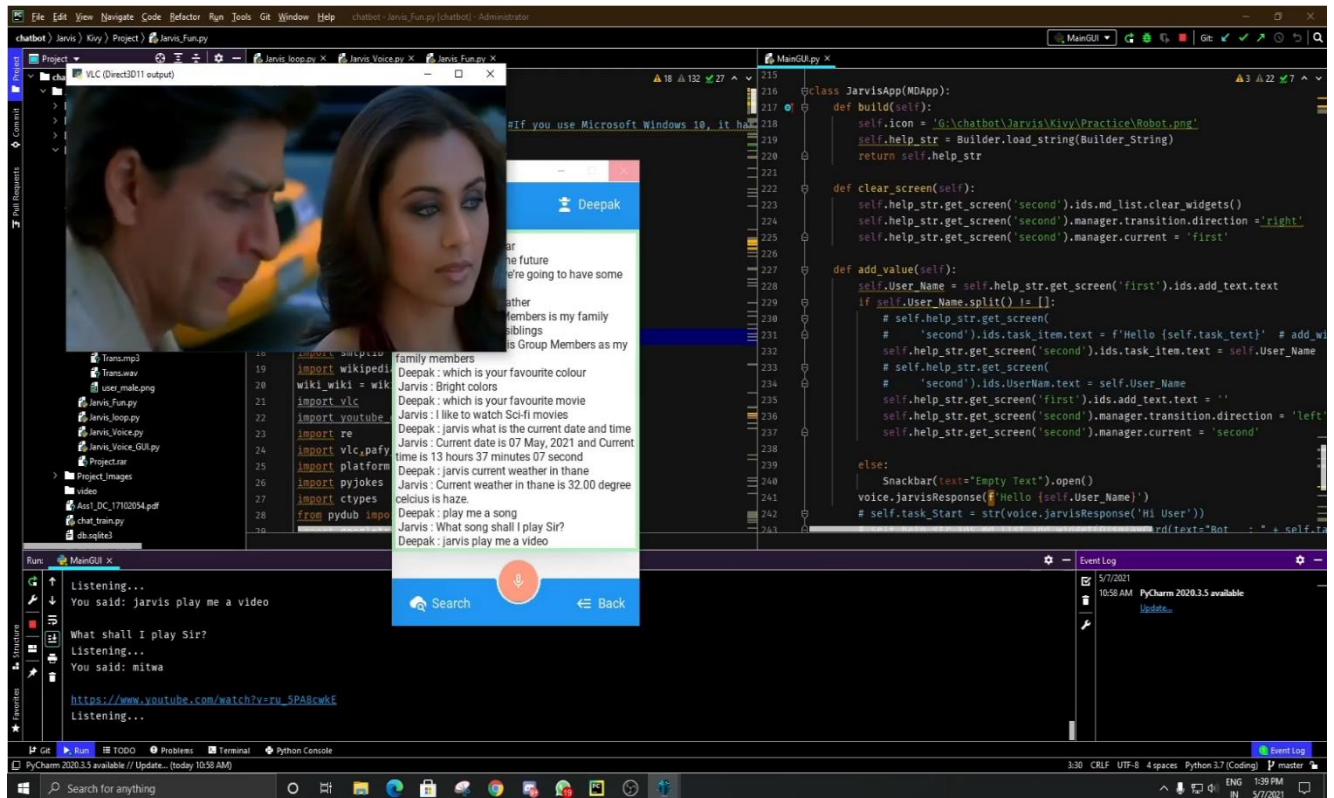


Figure 8: Result 2

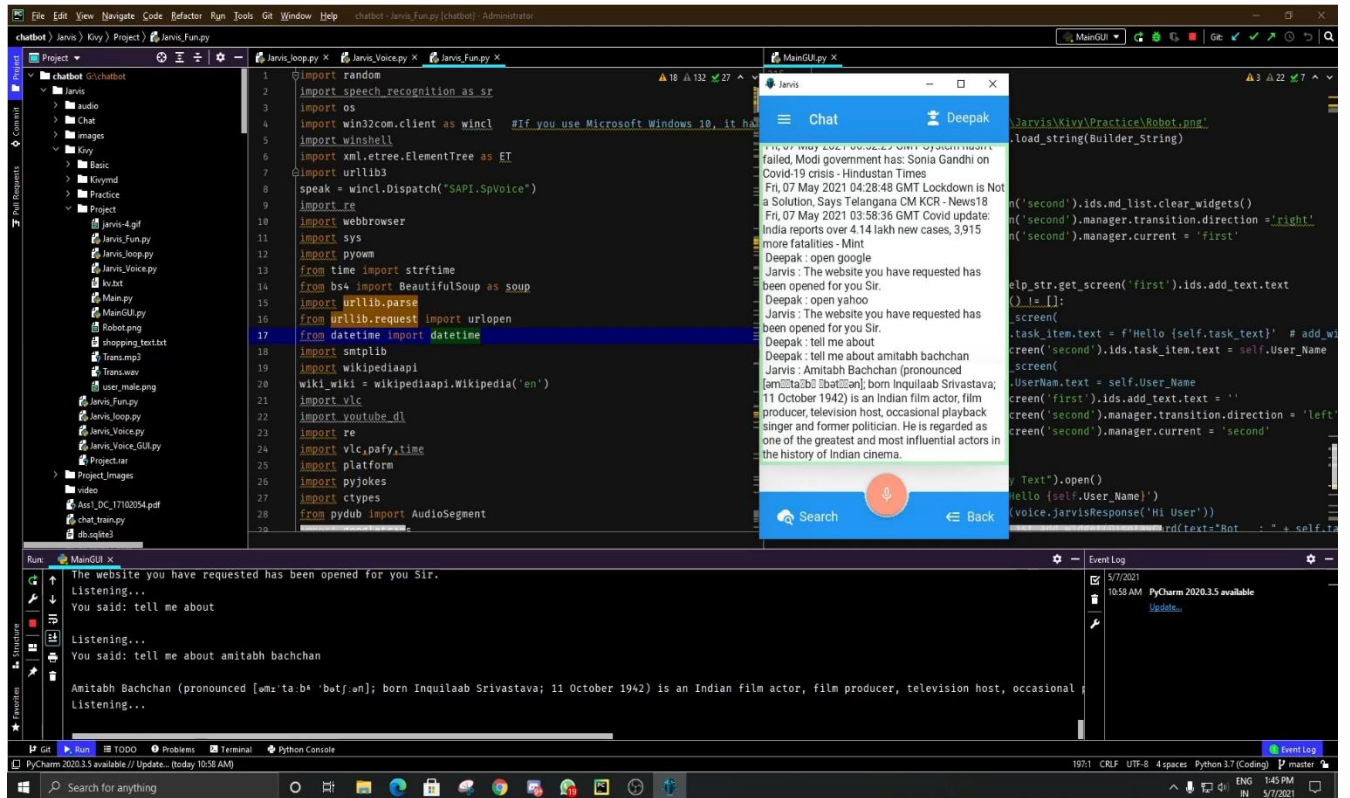


Figure 9: Result 3

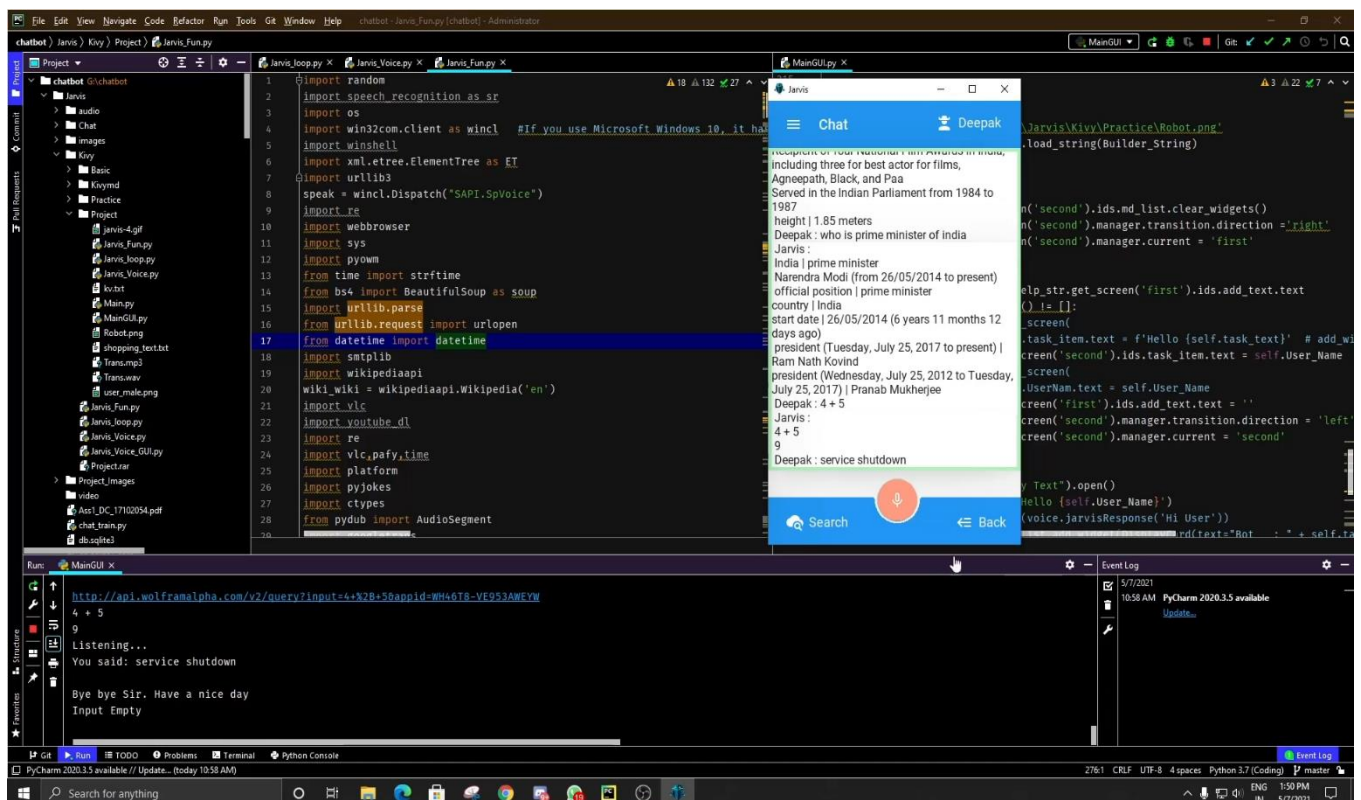


Figure 10: Result

## 5.Conclusion

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, translating word from one language to another language, accessing youtube videos, sending mail through voice and solving computational queries. We aim to make this project a complete User Interface based project and give user all its query on the very same User Interface.

## 6.References

- [1] V. Mitra, H. Franco, M. Graciarena and D. Vergyri, "Medium-duration modulation cepstral feature for robust speech recognition," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 1749-1753, doi: 10.1109/ICASSP.2014.6853898.
- [2] A. Acero, "An overview of text-to-speech synthesis," *2000 IEEE Workshop on Speech Coding. Proceedings. Meeting the Challenges of the New Millennium (Cat. No.00EX421)*, Delavan, WI, USA, 2000, pp. 1-, doi: 10.1109/SCFT.2000.878372.
- [3] T. Kim, "Short Research on Voice Control System Based on Artificial Intelligence Assistant," *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, Barcelona, Spain, 2020, pp. 1-2, doi: 10.1109/ICEIC49074.2020.9051160.
- [4] Jin He, Rui Yu, Xinsheng Wang and Lina Huang, "Validation of query expression based on Regular Expression," *2011 International Conference on Computer Science and Service System (CSSS)*, Nanjing, China, 2011, pp. 1879-1882, doi: 10.1109/CSSS.2011.5974145.
- [5] I. Budiselic, S. Sribljic and M. Popovic, "RegExpert: A Tool for Visualization of Regular Expressions," *EUROCON 2007 - The International Conference on "Computer as a Tool"*, Warsaw, Poland, 2007, pp. 23872389, doi: 10.1109/EURCON.2007.4400374.

- [6] D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 450- 454, doi: 10.1109/ICECA.2019.8822022.
- [7] S. S. Chowdhury, A. Talukdar, A. Mahmud and T. Rahman, "Domain Specific Intelligent Personal Assistant with Bilingual Voice Command Processing," *TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea (South), 2018, pp. 0731-0734, doi: 10.1109/TENCON.2018.8650203.
- [8] K. N., R. V., S. S. S. and D. R., "Intelligent Personal Assistant - Implementing Voice Commands enabling Speech Recognition," *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262279.
- [9] G. A. Triantafyllidis and M. G. Strintzis, "A least squares algorithm for efficient context- based adaptive arithmetic coding," *ISCAS2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, Sydney, NSW, Australia, 2001, pp. 169172 vol. 2, doi: 10.1109/ISCAS.2001.921034.
- [10] R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, "JARVIS: An interpretation of AIML with integration of gTTS and Python," *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies(ICICICT)*, Kannur, India, 2019, pp. 486489, doi: 10.1109/ICICICT46008.2019.8993344.
- [11] P. Bose, A. Malphak, U. Bansal and A. Harsola, "Digital assistant for the blind," *2017 2nd International Conference for Convergence in Technology (I2CT)*, Mumbai, 2017, pp. 1250-1253, doi: 10.1109/I2CT.2017.8226327.
- [12] M. R. Sultan, M. M. Hoque, F. U. Heeya, I. Ahmed, M. R. Ferdouse and S. M. A. Mubin, "Adrisya Sahayak: A Bangla Virtual Assistant for Visually Impaired," *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, DHAKA, Bangladesh, 2021, pp. 597-602, doi: 10.1109/ICREST51555.2021.9331080.

- [13] S. Kumari, Z. Naikwadi, A. Akole and P. Darshankar, "Enhancing College Chat Bot Assistant with the Help of Richer Human Computer Interaction and Speech Recognition," *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp.427433, doi:10.1109/ICESC48915.2020.9155951.

## **7.Bibliography**

Computational Speech Processing: Speech Analysis, Recognition, Understanding, Compression, Transmission, Coding, Synthesis; Text to Speech Systems, Speech to Tactile Displays, Speaker Identification, Prosody Processing: BIBLIOGRAPHY, by Conrad F. Sabourin, 1994, 2 volumes, 1187p, ISBN 2-921173-21-2, INFOLINGUA inc., P.O. Box 187 Snowdon, Montreal, H3X 3T4, Canada. See also: <http://gomer.mlink.net/infolingua.html>