**Speech Recognition And Language Translation Api**

*A Socially Relevant Project Report (19A05507) submitted to*

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTAPURAMU

*In partial fulfillment of the requirements for the award of the degree of*

# BACHELOR OF TECHNOLOGY

**In**

# COMPUTER SCIENCE AND ENGINEERING

##### Submitted by

B.V.MANOJ KUMAR(19HR1A0517) N.SATHISH KUMAR REDDY(19HR1A0568)

**A.SUMA(19HR1A0505) B. RETHASWI(19HR1A0519)**

*Under the esteemed guidance of*

# MRs N .V. NEERAJA QUALIFICATION M.TECH

ASST. PROF., CSE Department



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING MOTHER THERESA INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**Melumoi (Post), Palamaner-517408.**

**Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapuramu-515002 NAAC Accredited and An ISO 9001:2015 Certified Institution**

### 2021-2022

**MOTHER THERESA INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**AN ISO 9001:2015 CERTIFIED INSTITUTION**

**(Approved by AICTE, New Delhi and Affiliated to J.N.T.U.A., Anantapuramu) Melumoi (Post), Palamaner-517 408, Chitoor (Dist), A.P.**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

*This is to certify that the Socially Relevant Project (19A05507) Report entitled*

**SPEEECH RECOGNITION AND LANGUAGE TRANSLATION API**

*is the bonafide work done and*

*Submitted by*

B.V.MANOJKUMAR(19HR1A0517) N.SATHISH KUMAR REDDY (19HR1A0568)

**A.SUMA(19HR1A0505) B.RETHASWI(19HR1A0519)**

*In the Department of Computer Science and Engineering, Mother Theresa Institute of Engineering and Technology, Palamaner affiliated to J.N.T.U.A., Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science & Engineering during 2021-2022.*

**Submitted on:**

### Internal Guide HOD

**MRs. N .V . NEERAJA,M.Tech Dr. U. Kumaran**

**Asst. Prof. CSE dept HOD & Assoc. Prof.**

# ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual effort but on the guidance, encouragement and cooperation of intellectuals, elders and friends. We would like to take this opportunity to thank them all.

We feel ourselves honoured for placing our warm salutation to **THE MANAGEMENT** Mother Theresa Institute of Engineering and Technology, Palamaner, which gave us the opportunity to obtain a strong base in B. Tech and profound knowledge.

We express our sincere thanks to **Dr. M.LAKSHMIKANTHA REDDY, M.Tech, Ph.D,**

our beloved Principal for his encouragement and suggestions during our course of study.

With deep sense of gratitude we acknowledge **Dr. U. KUMARAN,** Head of the Dept., Computer Science& Engineering, for his valuable support and help in processing our Socially Relevant Project.

We also express sincere thanks to our Socially Relevant Project Coordinator

MRs N .V . NEERAJA, MTech, Asst.prof

Assistant Professor in Department of Computer Science and Engineering, for encouraging us in doing this Socially Relevant Project.

We whole-heartedly express gratitude and regards to our guide

**MRs N.V. NEERAJA, M.Tech, Asst.prof** in Department of Computer Science and Engineering, for constant support and encouragement.

Finally, we would like to express our sincere thanks to **Faculty Members** of CSE Department, and Lab Technicians, Friends & Family members, who all have motivated and helped us to do this Socially Relevant Project.

B.V.MANOJ KUMAR (19HR1A0517)

**N.SATHISH KUMAR REDDY (19HR1A0568)**

A.SUMA (19HR1A0505)

**B.RETHASWI (19HR1A0519)**

**(Signature)**

ABSTRACT

The main objective of project is to make language translation and voice typing much easier and faster in an efficient manner. Today communication play an important role, the language translation is more than just changing the words from one language to another. Translation builds bridges between communications. It allows you to experience phenomena that would betoo foreign and remote to grasp through your own communicational lens. This project also helps in improving voice typing which is used to translate the speech in standard and regional languages which reduces the effort of typing. Despite the fact that English has a far and wide reach today, the impact of local culture and language remains as strong as ever. With the growth of Internet and communication technology, it is relatively easier to reach audiences that are thousands of miles away purely on the back of effective translation. This in turn has resulted in a need for translation in diverse fields such as education, mass communication, science and technology, literature, tourism, religion, trade and business, etc

DEPARTMENT OF CSE MTIET,PALAMANER

### Table Of Contents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapter No** | **Chapter Name** | | | |
|  | **Abstract** | | | |
| 1 | **Introduction** | | | |
| 2 | **System requirements** | | | |
|  | 2.1 | Types of speech recognition | | |
| 2.2 | An overview o\f speech recognition | | |
| 2.3 | Weakness of speech recognition | | |
| 2.4 | Factors affecting speech recognition | | |
|  | 2.5 | The future of speech recognition | | |
| **3** | **Feasibility study** | | | |
|  | 3.1 | Technical study | | |
| 3.1.1 | Financial study | | |
| 3.2 | Operational study | | |
|  | 3.2.1 | Requirement analysis | |
| 3.2.2 | Existing system | |
| 3.2.3 | Tts flowchart | |
| 4 | **System Design And Implementation** | | | |
|  | 4.1 | General working mechanism of sppech to text conversion | | |
|  |  | 4.1.1 | | Functions of speech recognition |
|  | 4.1.2 | | speech dictation process |
| 5 | **Code Generation For speech recognition** | | | |
|  | 5.1 | **Code Generation For Home Page** | | |

DEPARTMENT OF CSE MTIET,PALAMANER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 5.2 | | **Code Generation For Speech to Text Html And Jscode** | |
|  | 5.3 | | **Code Generation For Music player** | |
|  |  | | 5.3.1 | **Code Generation For Music Player By Html** |
|  |  | | 5.3.2 | **Code Generation For Music Player By Css** |
|  |  | | 5.3.3 | **Code Generation For Music Plater Using Jscode** |
| 6 | **Conclusion** | | | |
|  | 6.1 | **Future Work** | | |

DEPARTMENT OF CSE MTIET,PALAMANER

### CHAPTER1:INTRODUCTION

##### 1.1Existing System (History)

* The concept of speech recognition started somewhere in 1940s, practically the first speech recognition program was appeared in 1953 at the bell labs, that was about recognition of a digit in a noise free environment.1940s and 1950s consider as the foundational period of the speech recognition technology, in this period work was done on the of the speech recognition that is automation and information theoretic models.In the 1960’s we were able to recognize small vocabularies (order of 10-100words) of isolated words , based on simple acoustic-phonetic properties of speech
* The key technologies that were developed during this decade were, filter banks and time normalization methods.In 1970s the medium vocabularies (order of 100-100 words) us template-based, pattern recognition methods were recognized.
* In 1980s large vocabularies (1000-unlimited) were used and speech recognitionproblems based on statistical, with a large range of networks for handling language structures were addressed. The key invention of this era were hidden mark model (HMM) and the stochastic language model , which together enabled powerful new methods for handling continuous speech recognition problem efficiently and with high performance.In 1990s the key technologies developed during this period were the methods forstochastic language understanding, statistical learning of acoustic and language models, and the methods for implementation of large vocabulary speech understanding systems.After the five decades of research, the speech recognition technology has finallyentered marketplace, benefiting the users in variety of ways. The challenge of designing a machine that truly functions like an intelligent human is still a major onegoing forward.

***1.2.*** *Project objective*

* Its application work in different areas
* Its implementation as a desktop Application
* This application as software that can be use for :
* Speech Recognition (convert the voice to text),Speech Generation , (convert the text to voice )
* Designing and development of an interactive user-friendly text editor , which allows the user to enter the text , manipulate text ,
* formatting text all by familiar commands .
* Developing software for speech recognition (speech to textconversion)
* Developing advanced technology incorporating these ideas.
* Development of a model that will compare the wave data with phoneme database and displaying the characters (sentences)on thescreen
* Speech recognition is a technology that able a computer to capture the words spokenby a human with a help of microphone (embedded in computeSpeech recognition Technology is one of the fast growing engineeringtechnologies.
* This project is designed and developed keeping that facto in mind , and a little effortis made to achieve this aim.
* It has a number of applications in different areas and provides potential benefits , Nearly 20% people of the world ae suffering from various disabilities ; many of themare blind or unable to use their hands effectively . The speech recognition system inthose particular cases provide a significant help to them , so that they can share information with people by operating computer through voice input .
* Consider the Thousands of people in world they are not able to use their hands making typing impossible. our project it for these people who can’t type ,and see
* ,even for those of us who are lazy and don’t feel like it Our project is capable to recognize the speech and convert the input audio into text; it also enables a user to perform operations such as (open , close ,exit, read, ……) program application and a file by providing voice input . example open **Word processing** ,**google chrome**
* ,**Notepad** and **calculator** …,,etc.
* In our project capable to read the text which is wrote by any one or the text which isentered by the user himself

**Glossary of term**s (key words):

* 1. ***ASR -*** *Automatic speech Recognition*
  2. **Dictation-** In which the user enters the data by reading directly to the computer.
  3. **AR** - Auto Regressive
  4. **ARMA** - Auto Regressive Moving Average
  5. **CD** - Cepstral Distortion
  6. DEPARTMENT OF CSE MTIET,PALAMANER
  7. **CDMA** - Code Division Multiple Access
  8. **CELP** - Code Excited Linear Prediction **DCT**

1. Discrete Cosine Transform
   1. **DFT** - Discrete Fourier Transform
2. Digital Signal Processing

**FEC** - Forward Error Correction

**FIR** - Finite Impulse Response

* 1. **GSM** - Global System for Mobile telecommunications
  2. **IIR** - Infinite Impulse Response
  3. **IDCT** - Inverse Discrete Cosine Transform**IDFT**

1. Inverse Discrete Fourier Transform

**LPC** - Linear Predictive Coding

* 1. **LSP** - Line Spectrum Pair
  2. **IMBE** - Improved Multi-Band Excitation
  3. **MBE** - Multi-Band Excitation
  4. **MSE** - Mean Square Error **NLP** - Non-Linear Pitch
  5. **PCM** - Pulse Code Modulation
  6. **PSTN** - Public Switched Telephone Network
  7. **RMS** - Root Mean Square **RPE** - Regular Pulse Excitation
  8. **SD** - Spectral Distortion
  9. **SEGSNR**- Segmental Signal to Noise Ratio
  10. **SNR** - Signal to Noise Ratio
  11. **VSELP** - Vector Sum Excited Linear Prediction **AMDF** - Averaged Magnitude Differentiate Function
  12. **F0** - Fundamental Frequency of Speech
  13. **STE** - Short Term Energy **ZCR** - Zero Crossing Rate **ITU** - Upper Energy threshold
  14. **ITL** - Lower Energy threshold
  15. **IZCT** - Zero Crossing Rate Threshold

1. **C-V** - Consonant Vowel
   1. **FFT** - Fast Fourier Transform
   2. **DFFT** - Discrete Fast Fourier Transform **STFT** - Short-Time Fourier Transform **MFCC** - Mel frequency cepstrum computation
   3. **DCT** - Discrete Cosine Transform

**Continuous speech:** When user speak in a more normal, fluid manner without having to pause between word, which is referred as continuous speech.

**Discrete speech:** when user speak with taking rest between each word then suchspeech is referred as discrete speech.

### Project Scope:

* + - * This project has the speech recognizing and speech synthesizing capabilities though it is not a complete replacement of what we call a notepad but still a good text editor to be used through voice
      * this software also can open windows based software such as notepad , google chrome and etc..

***Statement of the problem***

* + - Title of the present study was: “Speech Pattern Recognition for Speech To Text Conversion “
    - When user speaks any alphabets from the microphone the different patterns of the alphabets will be identified and it will be compared with the corresponding pattern stored in the standard phoneme databases and
    - corresponding highest matching alphabet of Gujarati language will be return in form of text on the screen.

### Rational of the study:

* The software helps the professional to manage their workload by dictation in speechlanguage.
* The software can operate transparently behind the application, benefiting users who are unfamiliar with speech recognition or who might become confused with multiple applications.
* Prepares such a standard file that can be used in another speech dictation or in any familiar editors.
* Enables end-users to mange their user files (i.e. opening , saving, backing up, restoring, renaming and deleting)
* Elimination of training to the software for every user. That means it is speaker independent speech dictation software. In addition, the software enhanced runtimes include support for customizable command and control functions, transcription, dictation playback.
* Speech can be saved in appropriate format, so that the speaker or a third partycan replay recorded speech to facilitate correction.
* Can switch between dictations and typing made without any extra efforts.
* The software uses its own changing fonts, and also supports other wellknown changing fonts.
* Hands-free computing as an alternative to the keyboard is impractical (i.e.small mobile devices, Auto PCs, or in mobile phones)
* Voice responses to message boxes and wizard screens can easily be designedinto an application.
* A more “human” computer, one user can talk to, may make educational and entertainment applications seem more friendly and realistic.

DEPARTMENT OF CSE MTIET,PALAMANER

* Streamlined access to application controls and large lists enables a user to speak any one item from a list or any command from a potentially huge set of commands without having to navigate through several dialog boxes or cascading menus.
* Speech activated macros let a user speak a natural word or phrase rather thanuse the keyboard or a command to activate a macro.
* Speech recognition offers game and edutainment developers the potential tobring their applications to a new level of play. It enhances the realism and fun in many computer games, it also provides a useful alternative to keyboard-based control, and voice commands provide new freedom for the user in any sort of application, from entertainment to office productivity.
* Applications that require users to keyboard paper-based data into the computer are good candidates for a speech recognition application. Readingdata directly to the computer is much easier for most users and can significantly speed up data entry.
* There are many situation in which hands are not available to issue commandsto a device such as it is natural alternative interferer to computers for peoplewith limited mobility in their arms & hands, or for those with sight limitations.
* Speech can be saved in appropriate format, so that the speaker or a third partycan replay recorded speech to facilitate correction.
* Can switch between dictations and typing made without any extra efforts.
* Hands-free computing as an alternative to the keyboard, or to allow the application to be used in environments where a keyboard is impractical (i.e. small mobile devices, Auto PCs, or in mobile phones)
* Voice responses to message boxes and wizard screens can easily be designed into an application.
* A more “human” computer, one user can talk to, may make educational and entertainment applications seem more friendly and realistic
* Streamlined access to application controls and large lists enables a user to speak any one item from a list or any command from a potentially huge set of commands without having to navigate through several dialog boxes or cascading menus.
* Speech activated macros let a user speak a natural word or phrase rather thanuse the keyboard or a command to activate a macro.
* Speech recognition offers game and edutainment developers the potential tobring their applications to a new level of play. It enhances the realism and fun in many computer games, it also provides a useful alternative to keyboard-based control, and voice commands provide new freedom for the user in any sort of application from entertainment to office productivity.

DEPARTMENT OF CSE MTIET,PALAMANER

* Applications that require users to keyboard paper-based data into the computer are good candidates for a speech recognition application. Reading data directly to the computer is much easier for most users and can significantly speed up data entry. Some recognizers can even handle spelling fairly well. If an application has fields with mutually exclusive data types (i.e. sex, age and city), the speech recognition engine can process the command and automatically determine which field to fill in.
* Document editing, in which one or both modes of speech recognition could be used to dramatically improve productivity. Dictation would allow users to dictate entire documents without typing. Command and control mode would allow users to modify formatting or change views without using the mouse or keyboard.
* The earliest use of Spoken Language Technologies was the reading machine, allowing blind people to read books. This field has now a whole array of technologies that also help hearing impaired children with interactive audio- visual software, and many other assistive approaches.

EPARTMENT OF CSE MTIET,PALAMANER

### Chapter 2:

### SYSTEMREQUIREMENTS

##### An overview of Speech Recognition

* Speech recognition is a technology that able a computer to capture the words spoken by a human with a help of microphone. These words are later on recognized by speech recognizer
* system outputs the recognized words. An ideal situation in the process of speech recognition is that, a speech recognition engine recognizes all words uttered by a human but, practically the performance of a speech recognition engine depends on number of factors. Vocabularies, multiple users and users and noisy environment are the major factorsthat are counted in as the depending factors for a speech recognition engine.

##### **2.1.** Types of speech recognition

* + Speech recognition systems can be divided into the number of classes based on their ability to recognize that words and list of words they have. A fewclasses of speech recognition are classified as under:
  + Isolated SpeechIsolated words usually involve a pause between two utterances; it doesn’t mean that it only accepts a single word but instead it requires one utterance at a time.
  + Connected words or connected speech is similar to isolated speech butallow separate utterances with minimal pause between them.
  + Continuous speech
  + Continuous speech allow the user to speak almost naturally , it is alsocalled the computer dictation.
  + Spontaneous Speech:
  + At a basic level, it can be thought of as speech that is natural sounding andnot rehearsed.
  + An ASR system with spontaneous speech ability should be able to handle a variety of
  + natural speech features such as words being run together, “ums” and “ahs”, and even slight stutters.

##### 1.3SpeechRecognitionweakness:

* + - Besides all these advantages and benefits, yet a hundred percent perfect speech recognition system is unable to be developed. There are number of factors that can reduces the accuracy and performance of a speech recognition program.Speech recognition process is easy for a human but it is a difficult talk for a machine
    - comparing with a human mind speech recognition programs seems less intelligent, this is due to that fact that a human mind is God gifted thing and the capability of thinking, understanding and reacting is natural, while for a computer program it is a complicated task, first it need to understand the spoken words with respect to their meanings, and it has to create a sufficient balance between the words, noise and spaces. A human has a built in capability of filtering the noise form
    - speech while a machine requires training, computer requires help for separating the speech sound from the other sounds.

DEPARTMENT OF CSE MTIET,PALAMANER

### Factors on the speech recognition:

* + 1. **Homonyms:** are the words that are differently spelled and have the different meaning but acquires the same meaning, for example “there” “their” “be” and “bee”, “cool” and “coal”. This is a challenge for computer machine to distinguish between such types of phrases that sound alike.
    2. **Overlapping speeches:** a second challenge in the process, is to understand the speech uttered by different users, current systems have a difficultyto separate simultaneous speeches form multiple users.
    3. **Noise factor:** the program requires hearing the words uttered by a human distinctly and clearly. Any extra sound can create interference, first you need to place system away form noisy environments and the nspeak clearly else the machine will confuse and will mix up the words.

### The future of speech recognition :

* Dictation speech recognition will gradually become accepted.
* Accuracy will become better and better.
* Microphone and sound systems will be designed to adapt more quickly to changing background noise levels, different environments, with better recognition of extraneous material to bediscarded.
* Greater use will be made of “intelligent systems” which will attempt to guess what the speaker intended to say, rather than whatwas actually said, as people often misspeak and make unintentionalmistakes.

### Methodology

* As an emerging technology , not all developers are familiar with speech recognition technology . While the basic functions of both speech synthesis and speech recognition takes only
* few minutes to understand (after all, most people learn to speak and listen by
* age two), there are subtle and powerful capabilities provided by computerized speech that developers will want to understand and utilize.
* An understanding of the capabilities and limitations of speech technology is also important for developers in making decisions about whether a particular applicationswill benefit from the use of speech input and output.

DEPARTMENT OF CSE MTIET,PALAMANER

#### System Requirements:

|  |  |  |
| --- | --- | --- |
| Component | Minimum | Recommended |
| CPU | 1.6 GHz | 2.53GHz |
| RAM | 2 GB | 4gb |
| Microphone | Mic | High quality microphones |
| Sound card | Sound card | Sound cards with very  clear signals |
|  |  |  |

CPU:-

* + Our Application depend on efficiency of CPU(central processing unit).
  + This is because a large amount of digital filtering and signal processing can take place in ASR(Automated Speech Recognition ).

**Chapter3: Feasibility Study :**

#### Feasibility Study has several aspects:

* 1. Technical Feasibility
  2. Financial Feasibility
  3. Operational Feasibility

Through these studies were obtained on the conclusions and proposals for the project

:

##### 3.1Technical Feasibility:

* There are many of components that can build our system computer devices which are use to implements the application(Speech Recognition System)

###### Human Components

* Programmers, Analyzers, Designers and etc...

###### Software Components

* Visual studio 2015: for build up our project, creates all the window forms application and designing an interfaces.
* MySQL: for managing the database (creates tables, store the data). Word processor: for write a project report.
* Programming language:The programming language is C SHARP(C#)
* Its easy for learning and its use for create windows forms application, itsalso a well-known and high-level programing language.Microsoft Speech **SDK** is one of the many tools that enable adeveloper to add **speech** capability in to an applications.C# is the open source language and run on Windows , Mac , and Linux. This language help you for developing the **windows** store application
* , **Android** apps, and **ios** apps. It can also be useful to build backend and middle-tire framework and libraries. It supports language interoperability it means that C# can access code written in any .NET compliant language .The C# can runs on a variety of computer platform so adeveloper can easily perform reusability of coding.C# supports operator overloading and pre-processor directive that will help for speech recognition grammar.With this language, we can easily handle speech recognitionevent. We can find freelance jobs online in this sector.The profits that the team can achieve after implementation the project ,in thebeginning ,
* will be a trial version and any user can use it for free. After improvingthe version there will be a product key that no one can use the application without it
* , and the application will be sold to users , every version will have different productkey .

|  |  |  |
| --- | --- | --- |
| * Component | * Minimum | * Recommended |
| * CPU | * 1.6 GHz | * 2.53GHz |
| * RAM | * 2 GB | * 4gb |

*3.3Operational Feasibility:*

* + - increasing recognition throughput in batch processing of speech data; and reducing recognition latency in realtime usage scenarios.
    - Improve Throughput: Allow batch processing of the speech recognition task to execute as efficiently as possible,thereby increasing the utility for multimedia search and retrieval.Batch speech transcription can be “embarrassingly parallel” by distributing different speech utterances to different machines. However, there is significant value in improving compute efficiency, which is increasingly relevant in today’senergy limited and form-factor limited devices and compute facilities.

**3.3.1Response Time (RT):**

* + This is widely used in the study of human speech recognition as a measure of relative processing difficulty at all levels ,including the sentence , word and phoneme levels. Information With the help of microphone audio is input to the system, the pc sound card produces the equivalent digital representation of received audio .An acoustic model is created by taking audio recordings of speech, and their text transcriptions, and using software to create statistical representations of the sounds that make up each word. It is used by a speech recognition engine to recognize speech .Speech is used in transactional applications to navigate around the application or to conduct a transaction.
  + For example, speech can be used to purchase stock, reserve an airline itinerary, or transfer bank account balances. It can also be used to follow links on the web or move from application to application on one's desktop. Most often, but not exclusively, this category of speech applications involves the use of a telephone. The user speaks into a phone, the signal is interpreted by a computer (not the phone), and an appropriate response is produced. A custom, application-specific vocabulary is usually used; this means that the system can ony "hear" the words in the vocabulary. This implies that the user can only speak what the system can "hear." These applications require careful attention to what the systemsays to the user since these prompts are the only way to cue the user as to which words can be used for a successful outcome.
  1. **Fundamentals to speech recognition**
* Speech recognition is basically the science of talking with the computer, andhaving it correctly recognized. To elaborate it we have to understand the following terms. Utterance when user say some things ,then this is an utterance in other words speakinga word or a combination of words that means something to computer is called an utterance.
* Utterances ate then sent to speech engine to be processed. Pronunciation a speech recognition engine uses a process word is its pronunciation, that represents what the speech engine thinks a word should sounds like. Words can themultiplepronunciations associated Grammar uses particular set of rules in order to define the words and phrases that are going to be recognized by speech engine, more concisely grammar define the domain with which the speech engine works. Grammar can be simple as list of words or flexible enough to support the various degrees of variations.
  + 1. Accuracy

The performance of the speech recognition system is measurable ;the ability of recognizer can be measured by calculating its accuracy. It is useful to identify an utterance.

* + 1. Vocabularies

Vocabularies are the list of words that can be recognized by the speech recognition engine. Generally the smaller vocabularies are easier to identify by a speech recognition engine, while a large listing of words are difficult task to be identified by engine.

* + 1. Training

Training can be used by the users who have difficulty of speaking or pronouncing certain words, speech recognition systems with training should be ableto adapt.

* 1. Tools
     1. **Visual studio 2015 (coding )**
     2. Office 2016 (word processor )for make documentation

## Structure analysis

**4.3. Speech Synthesis**

**Text to phoneme conversion**

## Prosody Analysis

**Wave form production**

Fig : 4.1 Speech Synthesis

A speech synthesizer converts written text into spoken language. Speechsynthesis is also referred to as text-to-speech (TTS) conversion.

The major steps in producing speech from text are as follows:

* ***Structure analysis***: process the input text to determine where paragraphs,sentences and other structures start and end. For most languages, punctuation and formatting data are used in this stage.
* ***Text pre-processing:*** analyze the input text for special constructs of the language. In English, special treatment is required for abbreviations, acronyms, dates, times, numbers, currency amounts, email addresses and many other forms. Other languages need special processing for these formsand most languages have other specialized requirements.

.The remaining steps convert the spoken text to speech.

* ***Text-to-phoneme conversion:*** convert each word to phonemes. Aphoneme is a basic unit of sound in a language. US English has around 45 phonemes including the consonant and vowel sounds. For example, "times" is spoken as four phonemes "t ay m s". Different languages have different sets of sounds (different phonemes). For example, Japanese has fewer phonemes includingsounds not found in English, such as "ts" in "tsunami".
* ***Prosody analysis:*** process the sentence structure, words and phonemes to determine appropriate prosody for the sentence. Prosody includes many of the features of speech other than the sounds of the words being spoken. This includesthe pitch (or melody), the timing (or rhythm), the pausing, the speaking rate, the emphasis on words and many other features. Correct prosody is important for making speech sound right and for correctly conveying the meaning of a sentence.
* ***Waveform production:*** finally, the phonemes and prosody information are used to produce the audio waveform for each sentence. Thereare many ways in which the speech can be produced from the phoneme and prosody information. Most current systems do it in one of two ways: concatenation of chunks of recorded human speech, or formant synthesis using signal processing techniques based on knowledge of how phonemes sound and how prosody affects those phonemes. The details of waveform

**CHAPTER4:**

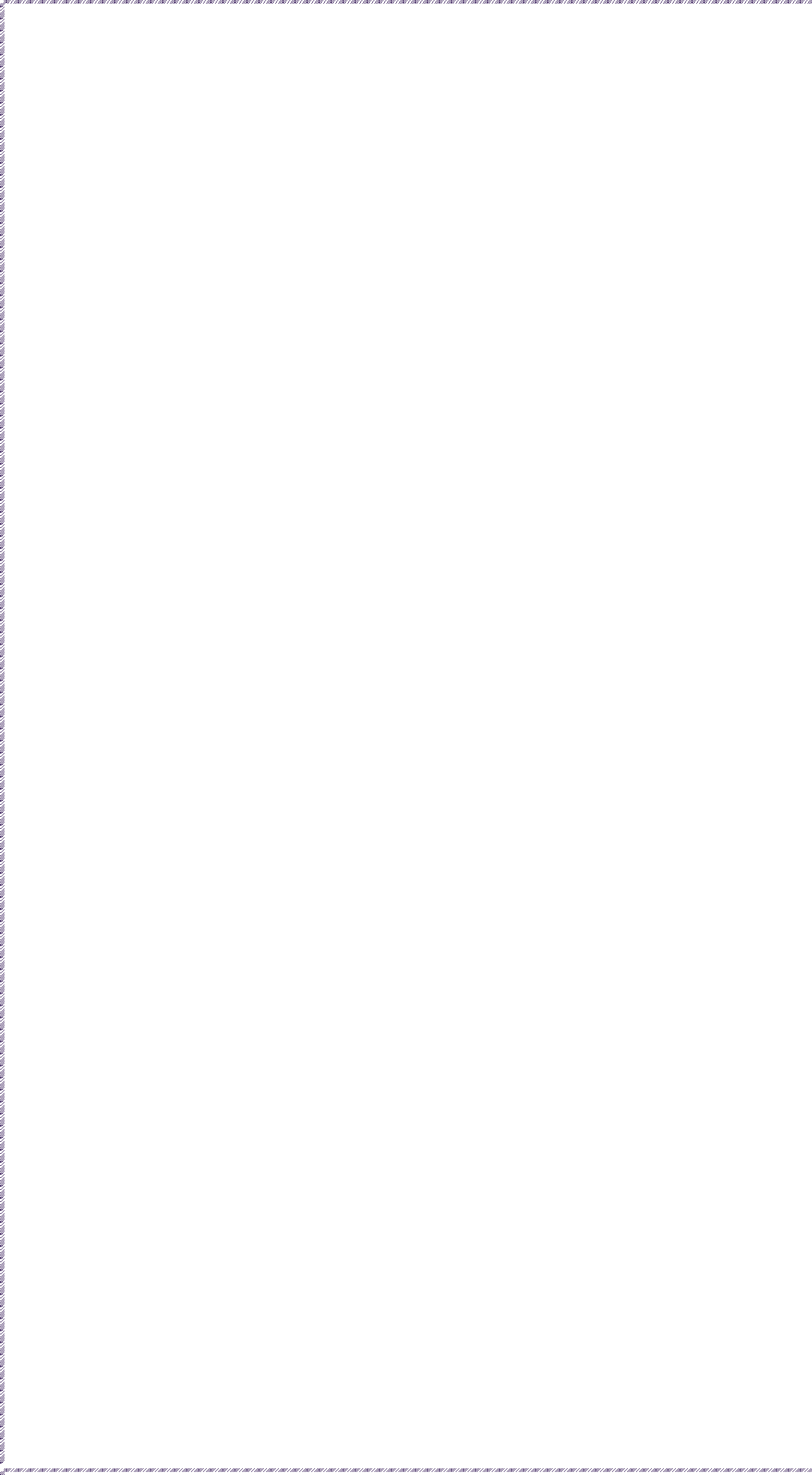
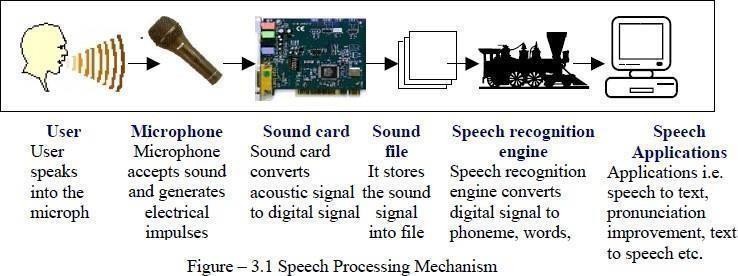
**SYSTEM DESIGN AND IMPLEMENTATION**

##### General mechanism of sppech detection:

* When one thinks about speaking to computers, the first image is usually speech recognition, the conversion of an acoustic signal to a stream of words. After many years of research, speech recognition technology is beginning to pass the threshold of practicality. The last decade has witnessed dramatic improvementin speech recognition technology, to the extent that high performance algorithms and systems are becoming available. Wide varieties of techniques with different levels of speech recognition are used to perform speech recognition. The speech recognition process is performed by a software component known as the speech recognition engine. The primary function of the speech recognition engine is to process spoken input word and translate it into text that an application understood. The application can be work i n two different mode Command and control mode some times referred as voice navigation and Dictation mode.
* In command and control mode the application can interpret the result of the recognition as command. This mode offers developers the easiest implementation ofa speech interface in an existing application. In this mode the grammar (or list of recognized words) can be limited to the list of available commands. This provides better accuracy and performance, and reduces the processing overhead required by the application. An example of a command and control application is one in which the caller says “open” “file”, and the application asks the name of the file to be opened.

**The entire speech recognize process is summarized as follow:**

* + Speech recognition starts with the digital sampling of speech.
  + The ASR program parses the noise, words and from the words it parse thephonemes, which are the smallest sound units.
  + The program database maps sounds to characters groups and converts intoan appropriate character group.

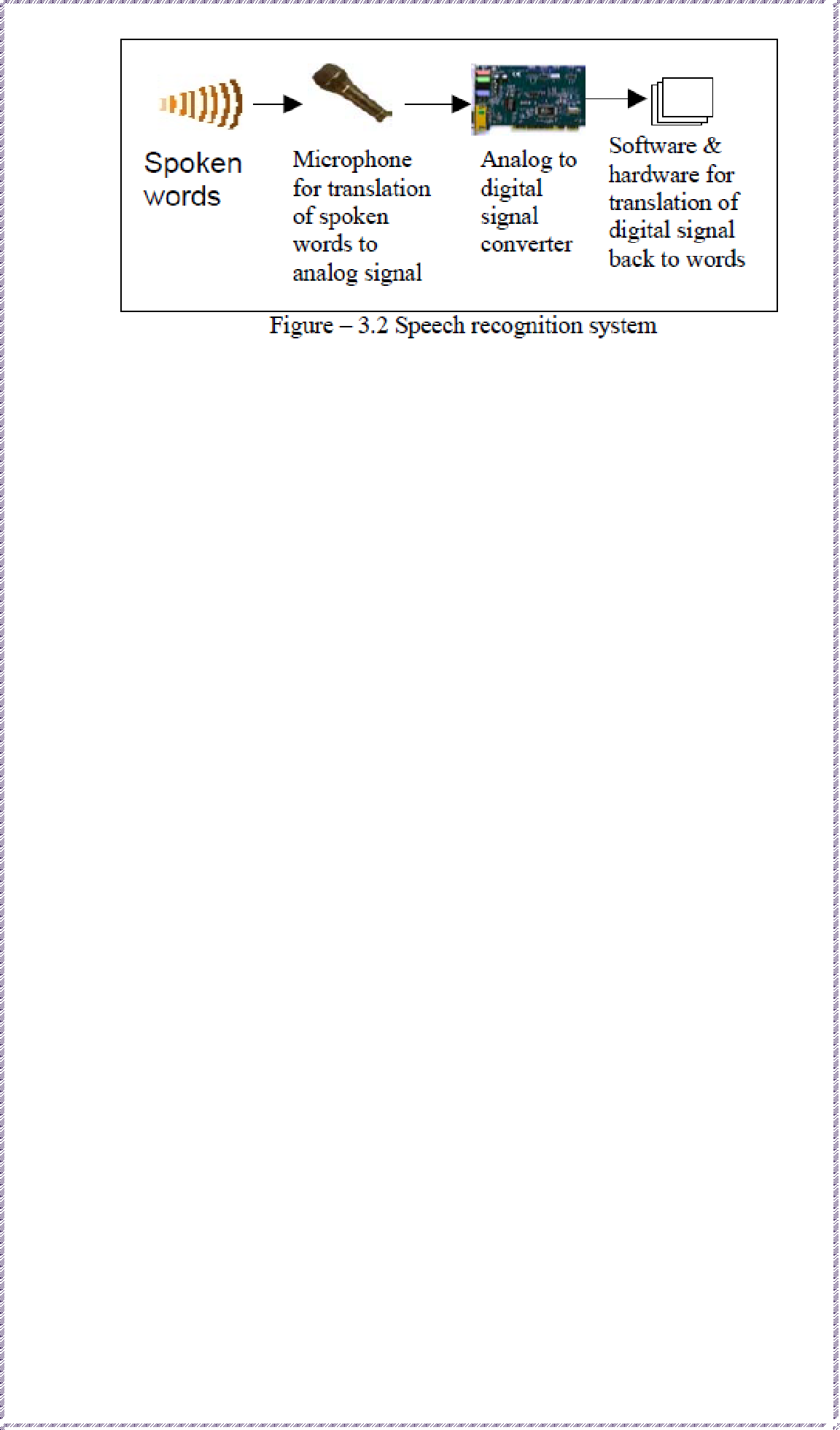
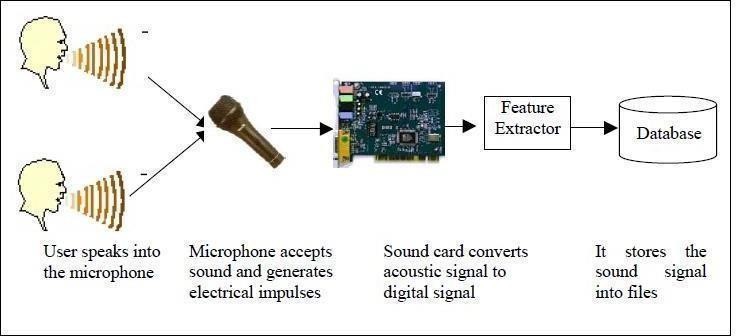


##### Functions of speech recognizer

* Functions of speech recognizer.
* Filters the raw signals into frequency bands.
* Cut the utterance into a fixed no. of segments.
* Average data for each band in each segment.
* Store this pattern with its name.
* Collect training set of about three repetitions of each patterns (utterance).
* Recognize unknown by comparing its pattern against all patterns in the training set and returning the name of the pattern closest to the unknown.

### A prototype model for speech recognition

* + - Speech recognition, or speech to text, involves capturing and digitizing the sound waves, converting them into basic language units or phonemes, constructing words from phonemes, and contextually analyzing the words to ensure correct spellingfor words that sound like. The figure-3.1 describes the on sight outline of the entire speech processing mechanism.
    - Normally a speech recognition system consists of three subsystems that includes microphone for translation of spoken words to analog signals, An analog-to-digital signal processor and software & hardware for translation of digital signal back to words.



### The process of conversion from speech to words is complex and varies slightly between systems. It consists of three steps :

* Feature extraction – Pre-processing of the speech signal, extracting the important features into feature vectors.
* Phoneme recognition– bases on a statistically trained phoneme model (HMM)the most likely sequence of phoneme is calculated.
* Word recognition – Based on statistically trained language model similar to the phoneme model, the most likely sequence of word

Fig: 5.3 Model for speech file preparation

***Speech dictation process :***

After the preparation of master database of features of Gujarati Alphabets, the researcher has proposed the dictation model from where the actual human- machine interaction starts in the form of speech dictation. The researcher has divided the model into five different steps

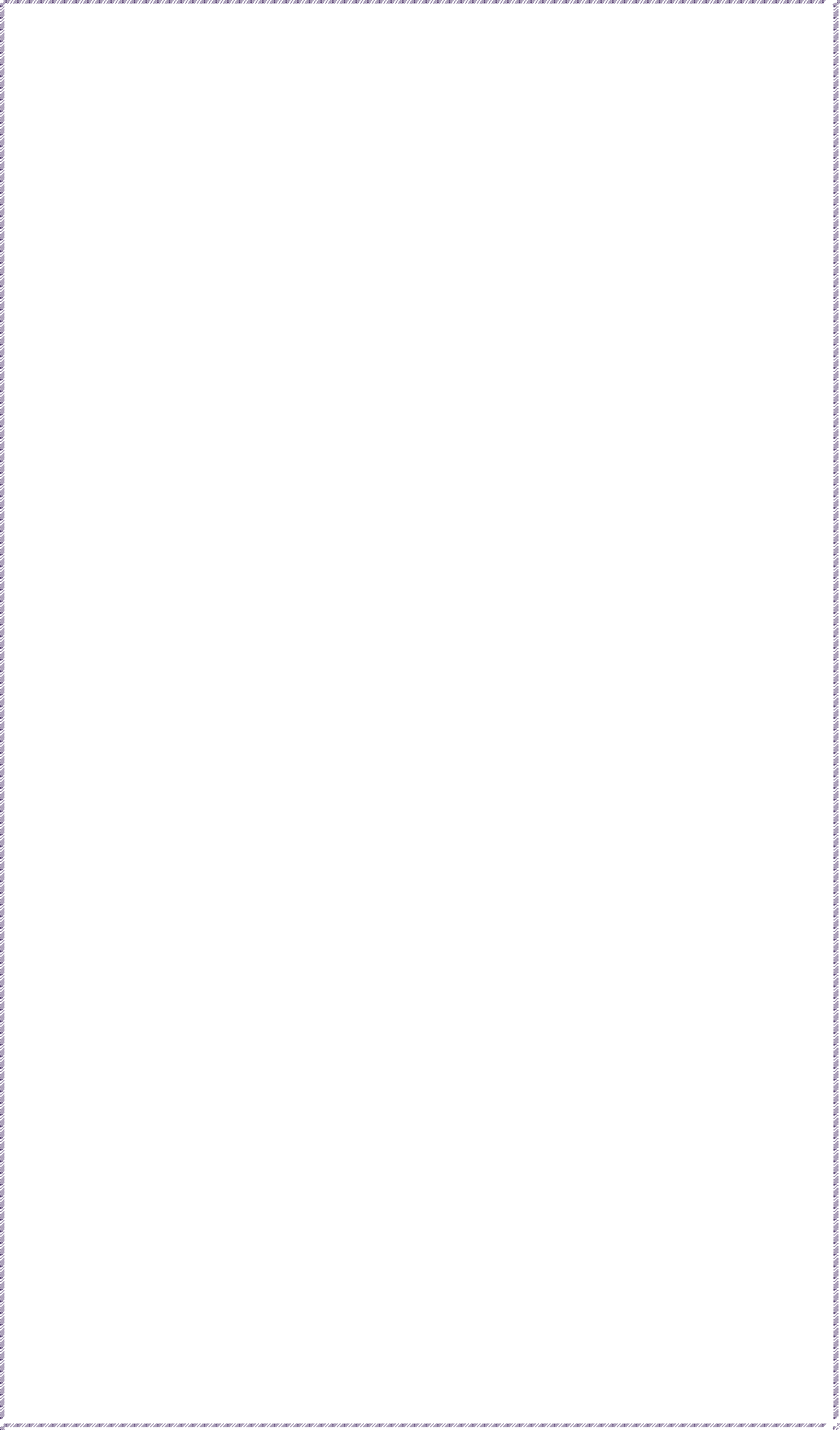
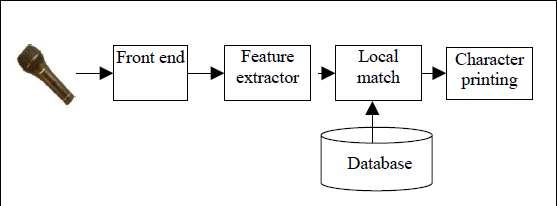
(1) Input acquition

Front end

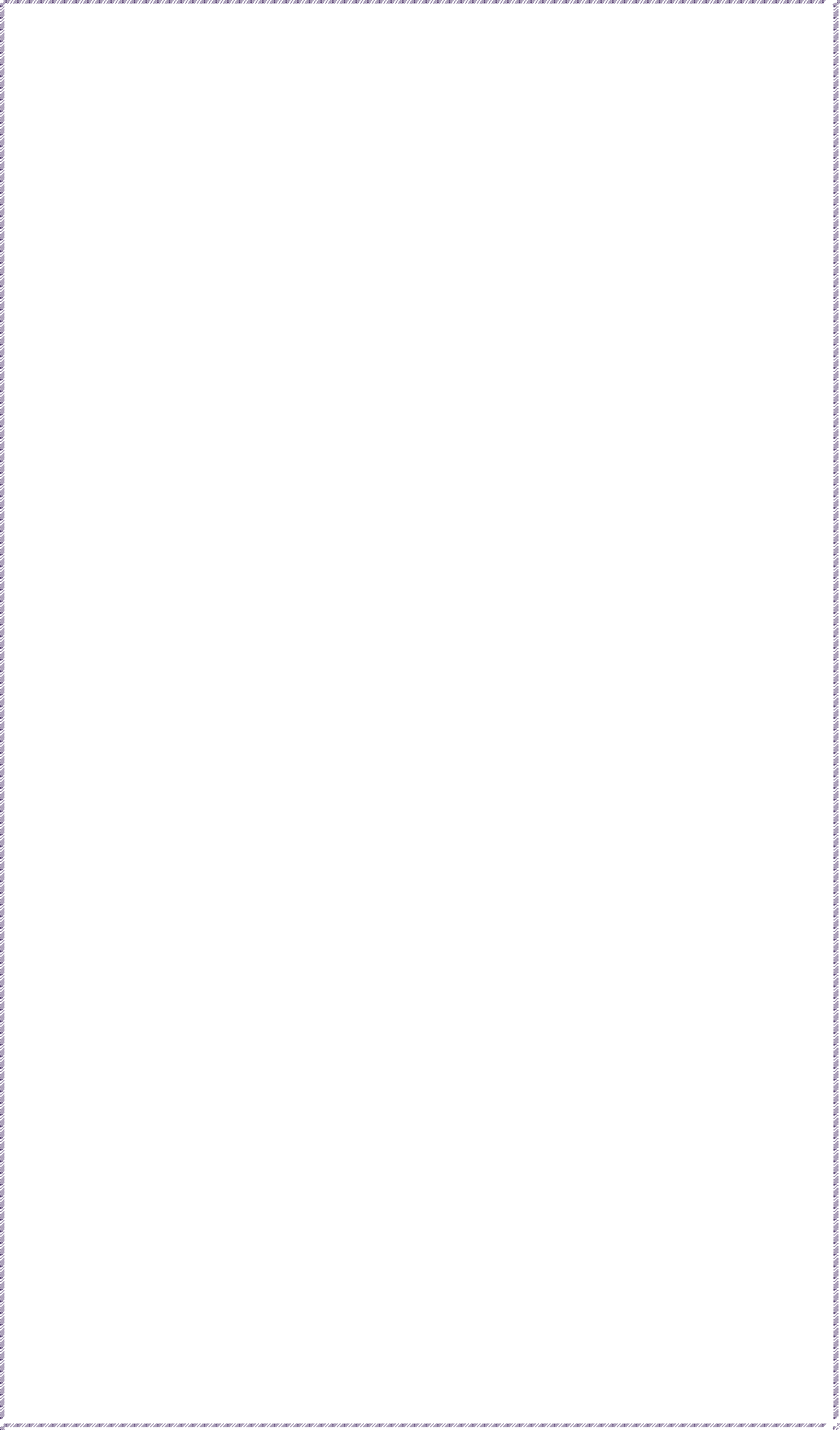
(3) Feature extractor

(4) local match and

(5) character printing.



*Fig: 5.4 speech dictation process*



## Flowchart of a speech recognition system

analysis

Robust processing

**Feature vector**

Speech Recognition

l

**Reference**

**result**

**resul**

**Chapter5:**

**Coding implementation for Speech Recognitition**

We divided the coding part into three parts Like creating individual webpages and connect themIn that we have

* 1. Home page 2.Text to speech

3. Music player with voice commands

# Home page code:

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" href="/Home Page/home.css">

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>demo</title>

</head>

<body>

<div class="conatiner-word">

<h1>Automatic Speech Recognition</h1>

<span></span>

<br>

<input type="text" id="transcript">

</div>

<div class="container-images">

<img src="/images/image 1.png" alt="" class="mic">

<img src="/images/image 2.png" alt="" class="pad">

<img src="/images/image 3.png" alt="" class="music" >

<img src="/images/image 5.png" alt="" class="book">

<img src="/images/image 6.png" alt="" class="msg">

</div>

</body>

<script src="//cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js"></script>

<script src='/Home Page/home.js'></script>

</html>

**Home page css:**

body {

background-color: black ; color: white;

font-family: cursive;

}

#transcript{

width: 8cm; height: 50px;

background-color: black; margin-top: 40px; border: transparent; color: aliceblue;

font-size: 3rem; font- family: cursive;

}

.conatiner-word{

position: relative;top: 40vh;

left: 5cm; width:

40vw;

}

.container-images{ width: 40vw; height: 50vh;

float: right; position: relative;top: 20vh; right: 30vh; border- radius: 50%;

}

.mic{

border-radius: 50%; position: relative;left: 8cm;

top: 4cm; width: 100px; cursor: pointer;

animation: size 3s ease-out;

}

@keyframes size {0%{ width: 2px;

} 100%{

width: 100px;

}

}

.pad{

position: relative;left: 6cm;

top: 1cm; width:

40px; cursor:

pointer;

animation: rotate 3s linear ;

}

@keyframes rotate {0%{

transform: rotate(0deg);width: 5px;

} 100%{

transform: rotate(360deg);width: 40px;

}

}

.music{

position: relative;left: 2.5cm;

top: 3cm; width:

40px; cursor: pointer;

animation: rotate 3s linear ;

}

.book{

position: relative;left: 6cm;

top: 3cm; width:

40px; cursor: pointer;

animation: rotate 3s linear ;

}

.msg{

position: relative;left: 2.5cm;

top: 5.5cm; width: 40px; cursor: pointer;

animation: rotate 3s linear ;

}

span::before{

content: ''; font-size: 30px;

font-family: cursive; animation: animate 6s infinite;

}

@keyframes animate { 0%{

content: 'Try Saying "Open Commands"';

} 50%{

content: 'Try Saying "Open Pad"';

} 100%{

content: 'Try Saying "Open Music"';

}

}

# Home page javascript:

const mic = document.querySelector('.mic'); mic.addEventListener('click', function (e) {

console.log('mic is clicked')

})

const pad = document.querySelector('.pad'); pad.addEventListener('click', function (e) {

window.open("/speech/speech.html","\_self")

})

const music = document.querySelector('.music'); music.addEventListener('click', function (e) { window.open("/music/music.html","\_self")

})

const book = document.querySelector('.book'); book.addEventListener('click', function (e) {

window.open("/commands/commands.html","\_self")

})

const msg = document.querySelector('.msg'); msg.addEventListener('click', function (e) {

window.open("/mail/mail.html")

})

if (annyang) {

var commands = {

'open music': function () { window.open("/music/music.html","\_self")

},

'open pad': function () { window.open("/speech/speech.html","\_self")

},

'open commands': function () { window.open("/commands/commands.html","\_self")

},

'open mail': function () { window.open("/mail/mail.html","\_self")

},

'\*tag': function (variable5) {

let transcript = document.getElementById("transcript") transcript.value = variable5

},

};

// Add our commands to annyang annyang.addCommands(commands);

// Start listening. You can call this here, or attach this call to an event, button, etc.

annyang.start();

}

**Speech to text HTML AND JS CODE:**

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Voice Controlled Notes App</title>

<meta name="description" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/shoelace-css/1.0.0- beta16/shoelace.css">

<link rel="stylesheet" href="/speech/speech.css">

</head>

<body>

<div class="container">

<h1>Speech recognition app</h1>

<div class="app">

<div class="input-single">

<textarea id="note-textarea" placeholder="Create a new note by typing or using voice recognition."

rows="6"></textarea>

</div>

<div class="all\_buttons">

<button id="start-record-btn" title="Start Recording">Start Recognition</button>

<button id="pause-record-btn" title="Pause Recording">Pause Recognition</button>

<button id="save-note-btn" title="Save Note">Save Note</button>

<p id="recording-instructions">Press the <strong>Start Recognition</strong> button and allow access.</p>

</div>

<h3>My Notes</h3>

<ul id="notes">

<li>

<p class="no-notes">You don't have any notes.</p>

</li>

</ul>

</div>

Select Voice: <select id='voiceList'></select> <br><br>

<button id='btnSpeak'>Speak!</button>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scri pt>

<script src="/speech/speech.js"></script>

<script src="//cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js"></script>

<script>

document.body.onkeyup = function(e){ if(e.keyCode == 32){

document.getElementById("pause-record-btn").click()

}

}

if (annyang) {

var commands = {

'Go back': function () {

window.open("/Home Page/home.html", "\_self")

},

'start': function () { document.getElementById("start-record-btn").click()

},

'pause': function () { document.getElementById("pause-record-btn").click()

},

'save': function () { document.getElementById("save-note-btn").click()

},

'speak':function(){ document.getElementById("btnSpeak").click()

}

}

// Add our commands to annyang annyang.addCommands(commands);

// Start listening. You can call this here, or attach this call to an event, button, etc.

annyang.start();

}

txtInput = document.querySelector('#note-textarea');

var varvoiceList = document.querySelector('#voiceList');var btnSpeak = document.querySelector('#btnSpeak');

var synth = window.speechSynthesis;var voices = [];

PopulateVoices(); if(speechSynthesis !== undefined){

speechSynthesis.onvoiceschanged = PopulateVoices;

}

btnSpeak.addEventListener('click', ()=> {

var toSpeak = new SpeechSynthesisUtterance(txtInput.value); var selectedVoiceName =

voiceList.selectedOptions[0].getAttribute('data-name'); voices.forEach((voice)=>{

if(voice.name === selectedVoiceName){ toSpeak.voice = voice;

}

});

synth.speak(toSpeak);

});

function PopulateVoices(){ voices = synth.getVoices();

var selectedIndex = voiceList.selectedIndex < 0 ? 0 : voiceList.selectedIndex;

voiceList.innerHTML = ''; voices.forEach((voice)=>{

var listItem = document.createElement('option'); listItem.textContent = voice.name; listItem.setAttribute('data-lang', voice.lang); listItem.setAttribute('data-name', voice.name); voiceList.appendChild(listItem);

});

voiceList.selectedIndex = selectedIndex;

}

</script>

</body>

</html>

# Speech to text css:

body{

background-color: #0e0d0d;

}

#note-textarea{ width: 700px; height: 550px;

position: relative; top: 2cm;

left: 14cm;

}

h1{

}

font-family: sans-serif; font-weight: 600; color: aliceblue;

text-align: center; margin-top: 40px;

.all\_buttons{

width: max-content; position: relative; bottom:14cm;

left: 0cm;

}

h3 ,#notes {

color: white; position: relative;bottom:14cm; left: 0cm;

}

#recording-instructions{ color: white;

}

#voiceList{

width: 30rem; position: absolute;bottom: 90%; left: 0rem;

}

#btnSpeak{

position: absolute; bottom: 80%;

left: 74%;top: 8cm;

}

.app{ width: 100%;

height: 200px;

}

# Speech to text js:

try {

var SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;

var recognition = new SpeechRecognition();

}

catch (e) { console.error(e);

$('.no-browser-support').show();

$('.app').hide();

}

var noteTextarea = $('#note-textarea');

var instructions = $('#recording-instructions'); var notesList = $('ul#notes');

var noteContent = '';

// Get all notes from previous sessions and display them. var notes = getAllNotes();

renderNotes(notes);

/\* Voice Recognition

-\*/

// If false, the recording will stop after a few seconds of silence.

// When true, the silence period is longer (about 15 seconds), recognition.continuous = true;

// This block is called every time the Speech APi captures a line. recognition.onresult = function (event) {

// event is a SpeechRecognitionEvent object.

// It holds all the lines we have captured so far.

// We only need the current one.var current = event.resultIndex;

// Get a transcript of what was said.

var transcript = event.results[current][0].transcript;

// Add the current transcript to the contents of our Note.

// There is a weird bug on mobile, where everything is repeated twice.

// There is no official solution so far so we have to handle an edge case.var mobileRepeatBug = (current == 1 && transcript == event.results[0][0].transcript);

if (!mobileRepeatBug) { noteContent += transcript; noteTextarea.val(noteContent);

}

};

recognition.onstart = function () {

instructions.text('Voice recognition activated. Try speaking into the microphone.');

}

recognition.onspeechend = function () {

instructions.text('You were quiet for a while so voice recognition turned itself off.');

annyang.start()

}

recognition.onerror = function (event) {if (event.error == 'no-speech') {

instructions.text('No speech was detected. Try again.'); annyang.start();

};

}

/\*

App buttons and input

-\*/

$('#start-record-btn').on('click', function (e) {if (noteContent.length) {

noteContent += ' ';

}

recognition.start(); annyang.abort()

});

$('#pause-record-btn').on('click', function (e) { recognition.stop();

instructions.text('Voice recognition paused.'); annyang.start()

});

// Sync the text inside the text area with the noteContent variable. noteTextarea.on('input', function () {

noteContent = $(this).val();

})

$('#save-note-btn').on('click', function (e) { recognition.stop();

if (!noteContent.length) {

instructions.text('Could not save empty note. Please add a message to your note.');

}

else {

// Save note to localStorage.

// The key is the dateTime with seconds, the value is the content of the note.

saveNote(new Date().toLocaleString(), noteContent);

// Reset variables and update UI. noteContent = ''; renderNotes(getAllNotes()); noteTextarea.val('');

instructions.text('Note saved successfully.');

}

})

notesList.on('click', function (e) { e.preventDefault();

var target = $(e.target);

// Listen to the selected note.

if (target.hasClass('listen-note')) {

var content = target.closest('.note').find('.content').text(); readOutLoud(content);

annyang.start()

}

// Delete note.

if (target.hasClass('delete-note')) {

var dateTime = target.siblings('.date').text(); deleteNote(dateTime); target.closest('.note').remove(); annyang.start()

}

});

/\*

Speech Synthesis

-\*/

function readOutLoud(message) {

var speech = new SpeechSynthesisUtterance();

// Set the text and voice attributes. speech.text = message;

speech.volume = 1;

speech.rate = 1;

speech.pitch = 1;

window.speechSynthesis.speak(speech);

}

/\*

Helper Functions

-\*/

function renderNotes(notes) { var html = '';

if (notes.length) { notes.forEach(function (note) {

html += `<li class="note">

<p class="header">

<span class="date">${note.date}</span>

<a href="#" class="listen-note" title="Listen to Note">Listen to Note</a>

}

else {

<a href="#" class="delete-note" title="Delete">Delete</a>

</p>

<p class="content">${note.content}</p>

</li>`;

});

html = '<li><p class="content">You don\'t have any notes

yet.</p></li>';

}

notesList.html(html);

}

function saveNote(dateTime, content) { localStorage.setItem('note-' + dateTime, content);

}

function getAllNotes() {var notes = [];

var key;

for (var i = 0; i < localStorage.length; i++) {key = localStorage.key(i);

console.log(i) console.log(key)

if (key.substring(0, 5) == 'note-') {notes.push({ date: key.replace('note-', ''),

content: localStorage.getItem(localStorage.key(i))

});

}

}

console.log(notes) return notes;

}

function deleteNote(dateTime) { localStorage.removeItem('note-' + dateTime);

}

# Music player HTML code:

<!DOCTYPE html>

<html>

<head>

<title>music player</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" href="music.css">

</head>

<body>

<div class="main">

<p id="logo"><i class="fa fa-music"></i>Music</p>

<!-- show\_song\_number -->

<div class="show\_song\_no">

<p id="present">1</p>

<p>/</p>

<p id="total">5</p>

</div>

<!--- left part --->

<div class="left">

<!--- song img --->

<img id="track\_image">

<div class="volume">

<p id="volume\_show">90</p>

<i class="fa fa-volume-up" aria-hidden="true" onclick="mute\_sound()" id="volume\_icon"></i>

<input type="range" min="0" max="100" value="90" onchange="volume\_change()" id="volume">

</div>

</div>

<!--- right part --->

<div class="right">

<!--- song title & artist name --->

<div class="song\_detail">

<p id="title">title.mp3</p>

<p id="artist">Artist name</p>

</div>

<!--- middle part --->

<div class="middle">

<button onclick="previous\_song()" id="pre"><i class="fa fa- step-backward"

aria-hidden="true"></i></button>

<button onclick="justplay()" id="play"><i class="fa fa-play" aria-hidden="true"></i></button>

<button onclick="next\_song()" id="next"><i class="fa fa-step- forward" aria-hidden="true"></i></button>

</div>

<!--- song duration part --->

<div class="duration">

<input type="range" min="0" max="100" value="0" id="duration\_slider" onchange="change\_duration()">

<button id="auto" onclick="autoplay\_switch()">Auto &nbsp;<i class="fa fa-circle-o-notch"

aria-hidden="true"></i></button>

</div>

</div>

</div>

<!--- Add javascript file --->

<script src="music.js"></script>

<script src="//cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js"></script>

<script>

if (annyang) {

var commands = {

'Go back': function () {

window.open("/Home Page/home.html", "\_self")

},

'Play': function playsong() {track.play();

play.innerHTML = '<i class="fa fa-pause" aria- hidden="true"></i>';

},

</html>

'stop': function pausesong() {track.pause();

play.innerHTML = '<i class="fa fa-play" aria- hidden="true"></i>';

},

'next song': function next\_song() {

if (index\_no < All\_song.length - 1) {index\_no += 1; load\_track(index\_no); playsong();

} else {

index\_no = 0; load\_track(index\_no);playsong();

}

},

'previous song': function previous\_song() {if (index\_no > 0) { index\_no -= 1; load\_track(index\_no);

playsong();

} else {

index\_no = All\_song.length;load\_track(index\_no); playsong();

}

},

};

// Add our commands to annyang annyang.addCommands(commands);

// Start listening. You can call this here, or attach this call to an event, button, etc.

annyang.start();

}

</script>

</body>

# Music player css code:

\*{

margin: 0;

padding: 0;

font-family: cursive;

}

body{

min-height: 100vh; display: grid; place- items: center;

background: rgba(0,0,0,0.5);

}

.main{

position: relative; min- height: 80vh; width:

80%; display: flex; align-items: center; justify-content: center;

background: #232427; border- radius: 8px;

box-shadow: inset 5px 5px 5px rgba(0,0,0,0.2),inset

-5px -5px 15px rgba(255,255,255,0.1), 5px 5px 15px rgba(0,0,0,0.3),

-5px -5px 15px rgba(255,255,255,0.1);

}

.main button{ padding: 10px 12px; margin: 0 10px;

}

.main #logo{

position: absolute;top: 10px;

left: 30px; font- size: 25px;color:

#ccc;

}

.main #logo i{ margin-right: 15px;

}

/\* left & right part \*/

.left{

width: 50%; display: flex; align-items: center;

justify-content: center;flex- direction: column;

}

/\* song image \*/

.left img{

height: 300px; width: 80%;

border-radius: 15px; object-fit: cover;

box-shadow: inset 5px 5px 5px rgba(0,0,0,0.2),inset

-5px -5px 15px rgba(255,255,255,0.1), 5px 5px 15px rgba(0,0,0,0.3),

-5px -5px 15px rgba(255,255,255,0.1);

padding: 5px;

}

/\* both range slider part \*/ input[type="range"] {

-webkit-appearance: none; width: 50%;

outline: none; height:

10px; margin: 0 15px; overflow: hidden; border- radius: 10px;

}

input[type="range"]::-webkit-slider-thumb{

-webkit-appearance: none; height: 10px;

width: 10px; background: #148F77;cursor: pointer;

box-shadow: -415px 0 0 400px #148F77;

}

.right input[type=range]{ width: 40%;

}

/\* volume part \*/

.left .volume{ margin-top: 25px;width: 80%; height: 30px; display: flex;

align-items: center; justify- content: center;color: #fff;

/\* border: 1px solid #fff;\*/

}

.volume input[type="range"] {flex: 1;

}

.left .volume p{

font-weight: bold;font- size: 15px;

}

.left .volume i{ cursor: pointer; padding: 8px 12px; background:

#148F77;

}

.left .volume i:hover{ background: rgba(245,245,245,0.1);

}

.volume #volume\_show{ padding: 8px 12px; margin: 0 5px 0 0;

background: rgba(245,245,245,0.1);

}

/\* right part \*/

.right{ width: 50%;

padding: 10px 0; display:

flex; align-items:

center;

flex-direction: column;

}

.right .middle{ width: 100%; display:

flex;

align-items: center; justify- content: center;

}

.right .middle button{ border: none; height:

70px; width: 70px; border-radius: 50%; display: flex;

align-items: center; justify- content: center;cursor:

pointer; outline: none; transition: 0.5s; background:

#232427;

box-shadow: inset 5px 5px 5px rgba(0,0,0,0.2),inset

-5px -5px 15px rgba(255,255,255,0.1), 5px 5px 15px rgba(0,0,0,0.3),

-5px -5px 10px rgba(255,255,255,0.1);

}

.song\_detail{

position: relative; width:

80%; overflow: hidden; margin-bottom: 6.5em;

/\* border: 1px solid #fff;\*/

}

.song\_detail #title{

text-transform: capitalize; color: #fff;

font-size: 35px;

}

.song\_detail #artist{

text-transform: capitalize; color: #fff;

font-size: 18px; margin-top: 5px;

}

.right .duration{ margin-top: 3em; position: relative; display: flex; align- items: center;

justify-content: center; width: 80%;

/\* border: 1px solid #fff;\*/

}

.duration input[type="range"] {flex: 1;

}

.right #auto{

font-size: 15px; text- align: center;cursor:

pointer; border: none; padding: 5px 7px; color:

#fff;

background: rgba(255,255,255,0.2); outline: none;

border-radius: 10px;

box-shadow: inset 2px 2px 5px rgba(0,0,0,0.2),inset

-2px -2px 5px rgba(255,255,255,0.1), 5px 5px 15px rgba(0,0,0,0.3),

-5px -5px 15px rgba(255,255,255,0.1);

}

#play{

background: #148F77;

}

.right button:hover{ background: #148F77;

}

.right i:before{ color: #fff; font-size:

20px;

}

.show\_song\_no{ position: absolute;top: 10px; right: 10px; width:

30px; height: 20px; display: flex;

align-items: center; justify- content: center;padding: 8px 12px; color: #fff;

border-radius: 5px;

background: rgba(255,255,255,0.2);

box-shadow: inset 2px 2px 5px rgba(0,0,0,0.2),inset

-2px -2px 5px rgba(255,255,255,0.1), 5px 5px 15px rgba(0,0,0,0.3),

-5px -5px 15px rgba(255,255,255,0.1);

}

.show\_song\_no p:nth-child(2){ margin: 0 5px;

}

/\*responsive\*/ @media(max-width: 700px){

.main{

min-height: 100vh;width: 100%;

flex-direction: column;

}

.right{

margin-top: 50px;width: 60%;

}

.right .duration{width: 90%;

}

.left{

margin-top: 5em;width: 60%;

}

.left img{

min-width: 90%;height: 180px;

}

.main #logo{ display: none;

}

.song\_detail{

position: absolute;top: 5px; left: 10px; width: 80%; height: 70px;

}

.song\_detail #title{ font- size: 1.8em;

}

}

@media(max-width: 500px){

.main{

min-height: 100vh;width: 100%;

flex-direction: column;

}

.right{

margin-top: 50px;width: 80%;

}

.left{

margin-top: 5em;width: 80%;

}

.left img{

min-width: 90%;height: 180px;

}

.main #logo{ display: none;

}

.song\_detail{

position: absolute;top: 5px; left: 10px; width: 80%; height: 70px;

}

.song\_detail #title{

font-size: 1.5em;

}

.song\_detail #artist{ font-size: 0.8em;

}

.right .middle button{ height: 62px; width: 62px;

}

}

# Music player js code:

let previous = document.querySelector('#pre'); let play = document.querySelector('#play'); let next = document.querySelector('#next'); let title = document.querySelector('#title');

let recent\_volume = document.querySelector('#volume'); let volume\_show = document.querySelector('#volume\_show'); let slider = document.querySelector('#duration\_slider');

let show\_duration = document.querySelector('#show\_duration'); let track\_image = document.querySelector('#track\_image'); let auto\_play = document.querySelector('#auto');

let present = document.querySelector('#present'); let total = document.querySelector('#total'); let artist = document.querySelector('#artist'); let timer;

let autoplay = 0; let index\_no = 0;

let Playing\_song = false;

//create a audio Element

let track = document.createElement('audio');

//All songs list let All\_song = [

{

name: "first song", path: "songs/1.mp3", img: "img1.jpg", singer: "1"

},

{

name: "second song", path: "songs/2.mp3",

singer: "2"

},

{

name: "third song", path: "songs/3.mp3",

img: "img3.jpg",singer:

"3"

},

{

name: "fourth song",path: "songs/4.mp3",img:

"img4.jpg", singer: "4"

},

{

name: "fifth song", path: "songs/5.mp3",img:

"img5.jpg", singer: "5"

}

];

// All functions

// function load the track function load\_track(index\_no) { clearInterval(timer); reset\_slider();

track.src = All\_song[index\_no].path; title.innerHTML = All\_song[index\_no].name; track\_image.src = All\_song[index\_no].img; artist.innerHTML = All\_song[index\_no].singer; track.load();

timer = setInterval(range\_slider, 1000); total.innerHTML = All\_song.length; present.innerHTML = index\_no + 1;

}

load\_track(index\_no);

//mute sound function function mute\_sound() { track.volume = 0;

volume.value = 0;

volume\_show.innerHTML = 0;

}

// checking.. the song is playing or not

function justplay() {

if (Playing\_song == false) { playsong();

} else { pausesong();

}

}

// reset song slider function reset\_slider() { slider.value = 0;

}

// play song function playsong() {

track.play(); Playing\_song = true;

play.innerHTML = '<i class="fa fa-pause" aria-hidden="true"></i>';

}

//pause song function pausesong() {

track.pause(); Playing\_song = false;

play.innerHTML = '<i class="fa fa-play" aria-hidden="true"></i>';

}

// next song

function next\_song() {

if (index\_no < All\_song.length - 1) {index\_no

+= 1; load\_track(index\_no); playsong();

} else { index\_no = 0;

load\_track(index\_no); playsong();

}

}

// previous song function previous\_song() {if

(index\_no > 0) { index\_no -= 1; load\_track(index\_no); playsong();

} else {

index\_no = All\_song.length; load\_track(index\_no); playsong();

}

}

// change volume

function volume\_change() { volume\_show.innerHTML = recent\_volume.value; track.volume = recent\_volume.value / 100;

}

// change slider position function change\_duration() {

slider\_position = track.duration \* (slider.value / 100); track.currentTime = slider\_position;

}

// autoplay function function autoplay\_switch() {

if (autoplay == 1) {autoplay

= 0;

auto\_play.style.background = "rgba(255,255,255,0.2)";

} else {

autoplay = 1; auto\_play.style.background = "#148F77";

}

}

function range\_slider() {let position = 0;

// update slider position

if (!isNaN(track.duration)) {

position = track.currentTime \* (100 / track.duration); slider.value = position;

}

// function will run when the song is overif (track.ended) {

play.innerHTML = '<i class="fa fa-play" aria-hidden="true"></i>';if (autoplay == 1) {

index\_no += 1; load\_track(index\_no);playsong();

}

### CONCLUSION

* This is the entire designing and development of a speech recognition. This is a speech recognition application. These days speech recognition is widely used and provides a way to real time chatting and communication .
* This is easy to use and gives fast instant response. Although, it has some disadvantages too. It has security risks and privacy concerns. Speech recognition has very bright future. A easy, flexible, simple vidchat application allows people to have a good conversation.
* Language translation is a whole new world of communication. It is fastest and latest form raising of business communications overall the globe. It is a quick communication channel to deliver short messages. It enables employees to communicate privately and quickly with each other.

### Future Work

* There is always a room for improvements in any software package, however good and efficient it may be done. But the most important things should be flexible to accept further modification .Right now we are just dealing with text communication. In future this software may be extended to include the following features
* Video chat: This will further enhance the feature of calling into video communication