Text to Speech Converter

Mini Project 1A Report

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering (Computer Engineering)
Second year Computer Engineering

by:

Chandani Raju Mourya	TU3F1920072
Rugved Milind Patil	TU3F1920085
Siddhi Dhanaji Dorage	TU3F1920062
Shweta Rajendra Saindane	TU3F1920108

Under the Guidance of

Prof. Vaishali Malpe

Department of Computer Engineering

TERNA ENGINEERING COLLEGE

Nerul (W), Navi Mumbai 400706

(University of Mumbai)

(2020-21)

Internal Approval Sheet

TERNA ENGINEERING COLLEGE, NERUL

Department of Computer Engineering

Academic Year 2020-21

CERTIFICATE

This is to certify that the mini project 1A entitled **"Text to Speech Converter"** is a bonafide work of

Chandani Raju Mourya	TU3F1920072
Rugved Milind Patil	TU3F1920085
Siddhi Dhanaji Dorage	TU3F1920062
Shweta Rajendra Saindane	TU3F1920108

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the Bachelor of Engineering (Computer Engineering).

Guide **Head of Department** **Principal**

Approval Sheet

Project Report Approval

This Mini Project 1A Report – entitled "**Text to Speech Converter**" by following students is approved for the degree of ***B.E. in "Computer Engineering"***.

Submitted by:

Chandani Raju Mourya	TU3F1920072
Rugved Milind Patil	TU3F1920085
Siddhi Dhanaji Dorage	TU3F1920062
Shweta Rajendra Saindane	TU3F1920108

Examiners Name & Signature:

1.----------

2.----------

Date: ----------

Place: ----------

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.

Chandani Raju Mourya	TU3F1920072	----------
Rugved Milind Patil	TU3F1920085	----------
Siddhi Dhanaji Dorage	TU3F1920062	----------
Shweta Rajendra Saindane	TU3F1920108	----------

Date: _____

Place: _____

Acknowledgement

We would like to express our sincere gratitude towards our guide **Prof.Vaishali Malpe**, Mini Project Coordinators **Prof. Ujwala Gaikwad, Prof. Nilesh Kulal, Prof. Dnyaneshwar Bavkar**, **Prof. Vishwajit Gaikwad** for their help, guidance and encouragement, they provided during the project development. This work would have not been possible without their valuable time, patience and motivation. We thank them for making our stint thoroughly pleasant and enriching. It was great learning and an honor being their student.

We are deeply thankful to **Dr. Archa Mire (H.O.D Computer Department)** and entire team in the Computer Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work.

We take the privilege to express our sincere thanks **to Dr. L. K. Ragha** our Principal for providing the encouragement and much support throughout our work.

Chandani Raju Mourya	TU3F1920072	----------
Rugved Milind Patil	TU3F1920085	----------
Siddhi Dhanaji Dorage	TU3F1920062	----------
Shweta Rajendra Saindane	TU3F1920108	----------

Date: _____

Place: _____

Abstract

Text-to-speech (TTS) is the generation of synthesized speech from text. Text-To-Speech (TTS) is a technology that converts a written text into human understandable voice. A TTS synthesizer is a computer based system that can be able to read any text aloud that is given through standard input devices. Our goal is to make synthesized speech as intelligible, natural and pleasant to listen, as human speech. Speech is the primary means of communication between people. During synthesis very small segments of recorded human speech are concatenated together to produce the synthesized speech. The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. A text-to-speech synthesizer allows people with visual impairments and reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s. Recent progress in speech synthesis has produced synthesizers with very high intelligibility but the sound quality and naturalness still remain a major problem. However, the quality of present products has reached an adequate level for several applications, such as multimedia and telecommunications. Here, we developed a useful text-to-speech synthesizer in the form of a simple software that converts inputted text into synthesized speech and reads out to the user. The development of a text to speech synthesizer will be of great help to people with visual impairment and make making through large volume of text easier

The following thesis presents a brief overview of the main text-to-speech synthesis problems, and the initial work done in building a TTS in English.

Table of Contents

List of Figures

CHAPTER – 1

Introduction

Language is the ability to communicate one's thoughts by means of a set of signs (text), gestures, and sounds. It is a distinctive feature of human beings, who are the only creatures to use such a system. Speech is the oldest means of communication between people and it is also the most widely used. Speech is most widely used for communication between people. 'Speech synthesis' also called 'Text to speech synthesis' is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer and can be implemented in software.. Text to speech (TTS) conversion is the process of converting information stored as a data or text into speech. It is useful for blind people as audio reading device. There are many speech synthesizers using complex neural network design. The important qualities of speech synthesis systems are naturalness and intelligibility. Naturalness describes the output speech sounds similar to human speech and intelligibility is which the output speech sound is understood.

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in
software or hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. Text-to-speech (TTS) convention transforms linguistic information stored as data or text into speech

Text-to-speech synthesis -TTS - is the automatic conversion of a text into speech that resembles, as closely as possible, a native speaker of the language reading that text. Text-to speech synthesizer (TTS) is the technology which lets computer speak to you. The TTS system gets the text as the input and then a computer algorithm which called TTS engine analyses the text, pre-processes the text and synthesizes the speech with some mathematical models. The TTS engine usually generates sound data in an audio format as the output.

1.1Aim and Objective of Project:

The main aim is to create a Text to speech converter to help visually impaired people to enter the text and listen it out loud without any difficulty.

The main objectives of this project are

- To describe the problems in creating speech for e-learning course
- To search the Text–to–Speech software tools on the market
- To identify Text–to–Speech software needs of e-learning program
- To implement the comparison testing
- To recommend the most appropriate Text–to–Speech software tools for e-learning development.
- To make available a software for visual impaired people to help them in listening the text which they are unable to read or have problem in reading.
- To provide a human like many voices in different rate, pitch and volume .

The practical part of the project was a constructive researching and testing of the text–to–speech software tool features that would fulfill the requirements of Any e-learning course development.

1.2 Scope:-

The scope of this project is to help the user to listen the text out loud free without any cost and without any difficulties. We have seen that there must be some delay in generating the voice We can think of a method which recognize it and removes it automatically. In future we are thinking of adding the text highlighter so the user can see what text is being read by computer and also will work on grammer check and word correction.

1.3 Significance of the report:-

This project has theoretical, practical, and methodological significance:
The speech synthesizer will be very useful to any researcher who may wish to venture into the "Impact of using
Computer speech program for brain enhancement and assimilation process in human beings".
This text-to-speech synthesizing system will enable the semi-illiterates assess and read through electronic
documents, thus bridging the digital divide. The technology will also find applicability in systems such as
banking, telecommunications (Automatic system voice output), transport, Internet portals, accessing PC,
emailing, administrative and public services, cultural centres and many others. The system will be very useful to
computer manufacturers and software developers as they will have a speech synthesis engine in their
applications.

Advantages:

- The system is helpful for persons having learning disabilities or visually challenged.
- Prevents eye from strain, and user can sit and listen comfortably.
- Saves time especially while driving, exercising.
- Easy to use.
- Help improving spelling, reading, writing skills.

Disadvantages:

- Not natural sounding.
- Cannot read symbols.

1.4 Organization of the report:-

Chapter 1 gives a brief overview about the Introduction, aim and objective for developing this project.

Chapter 2 of the report includes the literature review and literature survey on the existing system.

Chapter 3 The problem definition tells us about the expected outcome of the project for the application.

Chapter 4 shows the block diagrams of text to speech converter and contains information about design and implementation.

Chapter 5 contains implementation details and working of the system.

Chapter 6 shows the methodology and result of experiments.

Chapter 7 contains the screenshots of software.

Chapter 8 contains conclusion and reference of the project.

Chapter 2:

2.1 Literature Review

In this review paper we have analysed the existing system for speech recognition, and speech to text conversion.

A. SPEECH RECOGNITION

Speech Recognition is the ability of machine/program to identify words and phrases in spoken language and convert them into machine-readable format. Speech Recognition Systems can be classified on basis of the following parameters:

• **Speaker:** All speakers have a different kind of voice. The models hence are either designed for a specific speaker or an independent speaker.

• **Vocal Sound:** The way the speaker speaks also plays a role in speech recognition. Some models can recognize either single utterances or separate utterance with a pause in between.

• **Vocabulary:** The size of the vocabulary plays an important role in determining the complexity, performance, and precision of the system.

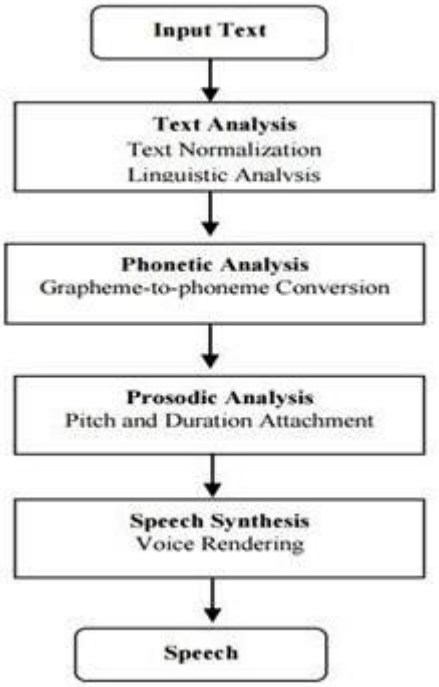


Fig 2.1 Text to speech conversion

A. TEXT TO SPEECH CONVERSION

Text-To-Speech is a process in which input text is first analysed, then processed and understood, and then the text is converted to digital audio and then spoken. Above Figure shows the block diagram of TTS. The figure shows all the steps involved in the text to speech

conversion but the main phases of TTS systems are:

• **Text Processing**: The input text is analysed, normalized (handles acronyms and abbreviation and match the text) and transcribed into phonetic or linguistic representation.
• **Speech Synthesis:** Some of the speech synthesis techniques are

i) Articulator Synthesis: Uses mechanical and acoustic model for speech generation. It produces intelligible synthetic speech but it is far from natural sound and hence not widely used.

ii) Formant Synthesis: In this system, representation of individual speech segments are stored on a parametric basis. There are two basic structures in formant synthesis, parallel and cascade, but for better performance, some kind of combination of these 2 structures is used. A cascade formant synthesizer consists of band-pass resonators connected in series. The output of each formant resonator is applied to the input of the successive one. The cascade structure needs only formant frequencies as control information. A parallel formant synthesizer consists

of resonators connected in parallel. The excitation signal is applied to all formants simultaneously and their outputs are summed. [5]

iii) Concatenative Synthesis: This technique synthesizes sound by concatenating short samples of sound called units. It is used in speech synthesis to generate user specific sequence of sound from a database built from the recording of other sequences. Units for Concatenative synthesis are [5]: Phone- a single unit of sound; Diphone- is defined as the signal from either midpoint of a phone or point of least change within the phone to the similar point in the next phone; Triphone- is a section of the signal taking in a sequence going from middle of a phone completely through the next one to the middle of a third.

2.1 Literature Survey

The present speech synthesis systems can be successfully used for a wide range of diverse purposes. However, there are serious and important limitations in using various synthesizers. Many of these problems can be identified and resolved. The aim of this paper is to present the current state of development of speech synthesis systems and to examine their drawbacks and limitations

he system is very time consuming as it requires huge databases and hard-coding of combination to form these words. As a result speech synthesis consumes more processing power.

➡The resulting speech is less than natural and emotionless. This is because it is impossible to get audio recordings of all possible words spoken in all the possible combinations of emotions, prosody, stress etc.

➡Pronunciation analysis from written text is a major concern.

➡It is difficult to build a perfect system.

➡Filtering background noise is a task which can even be difficult for humans to accomplish

But in this modern era,TTS helps improving word recognition and it makes students more likely to identify and fix their mistakes.

While some still maintain that audiobooks -or in general texts read by voice actors- are preferable to computer-generated voices, there is strong evidence supporting the claim that TTS can be an incredibly valuable tool for students of all ages.

The level of advancement reached by text-to-speech technology is enough that computer-generated voices can be engaging in ways that only a few years ago seemed unthinkable.

Chapter 3

Problem Statement

The importance of texts cannot be overemphasized. Hardly can anyone pass a message without including one form of text or the other. This is a problem for the visually impaired. They find it hard to read through the texts especially when the font-size is small. This has led to the development of a text to speech conversion system. For those with learning disabilities, some in literary levels, they often get frustrated trying to browse the internet because so much of it is in text form.

Also in some already developed speech synthesizers, the problem area in speech synthesis is very wide. There are several problems in text pre-processing, such as numerals, abbreviations, and acronyms. This system will help solve the problems by using well written synthesis algorithm for the conversion.

Even for people with the visual capability to read, the process can often cause too much strain to be of any use or enjoyment. With text to speech, people with visual impairment can take in all manner of content in comfort instead of strain.

We build text–to–speech solutions to create the speech for e-learning courses. Hiring native speakers to record the speech is not a good option.

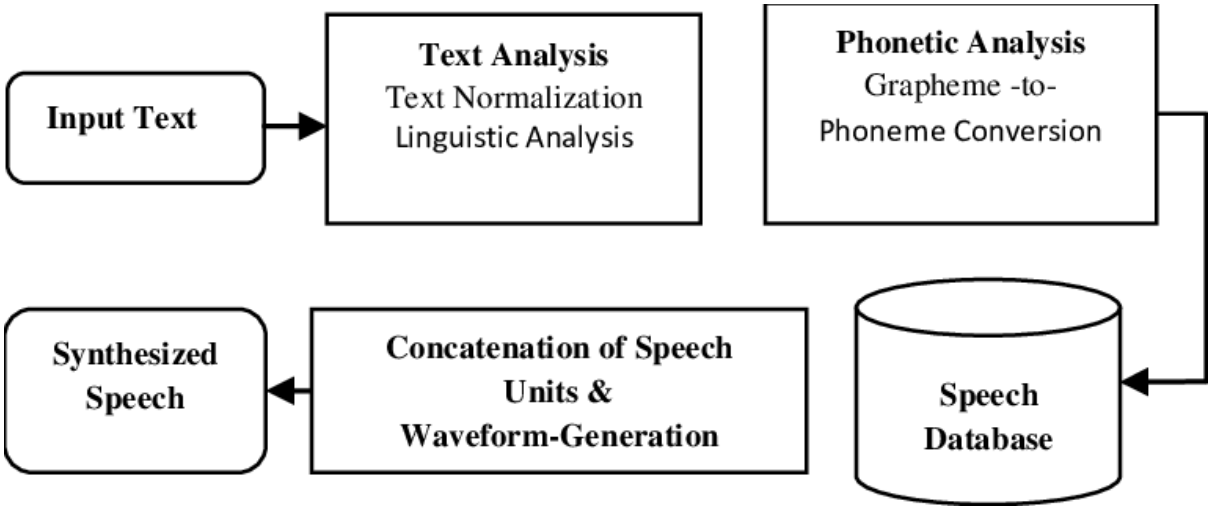
The text–to–speech software tool is preferred for creating the speech for e-learning courses. In the past year, {NAME} TTS was used as the main tool in e-learning course development. However, it didn't fulfill the e-learning developers' needs, especially in user interface and voice quality.

It appears more text–to–speech software tools are continuously produced with new solutions and improvements for speech synthesis technology. The different functionalities of the software tools as well as increasing price competition make it important to compare the available text–to–speech tools.

Hence, the success of comparison and selection of text–to–speech product will promote the quality of e-learning.

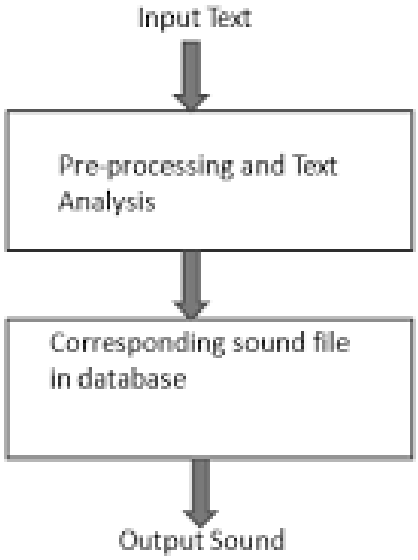
Chapter 4

Design and implementation



4.1 Text to speech block diagram

The above diagram shows the conversion of Text into speech in block diagram format.
When user enters the text in textbox the computer starts to analyse the text The two analyzation takes place one is text analysis and the other is phonetic analysis.
Then the computer goes to speech database Where the voices are stored. Then The concatenation of speech takes place and waveforms generates hence user listen the synthesized speech as an output.



4.2 Text to speech simple diagram
The above diagram shows the same process in simple steps .

4.1 Hardware and Software Requirements

Software Requirements

- Operating system: Windows XP/ Fedora core-I
- Software: Laravel, codepen and VS code
- Language used: HTML, CSS , JAVASCRIPT, php

Hardware Requirements

· Any x86 class processor
· 32 MB RAM
· 1 GB Hard Disk Space.
· Speaker connected to the computer.

Chapter 5

Implementation details

Our software is called the TextToSpeech Converter, a simple software with the text to speech functionality. The system was developed using JAVASCRIPT language. JAVASCRIPT is used since it is object oriented language.
We implement the web API called freeTTS TextToSpeech Converter (TTSC) converts text to speech either by typing the text into the text field provided or by coping from an external document in the local machine and then pasting it in the text field provided in the application.

Word to voice conversion:-
Let us start text to speech synthesis with a simple word to voice conversion.
All the data is stored in the form of a matrix. For every element read, corresponding wave file is played so as to output the sound of that character. we can read as big file as possible but only character wise.

Testing:
A text with few characters is created. then opened and read. For every character read, corresponding voice with different rate pitch and volume is played.

When we are successful in playing a word correctly, then stream of few words can be easily played. And the results obtained were quite satisfactory.

ALGORITHM
STEP1: Create a function and add speech synthesis in JAVASCRIPT.
STEP2: Type the Text in the Text Field.
STEP3: Click the “Speak It” Button.
STEP4: Computer will Read the Text written

5.1 Implementation Code

1. Home Page

HTML CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" type="text/css" href="C:\Users\CHZ\Desktop\study\textspeech.css">


  <script src="C:\Users\CHZ\Desktop\study\\textspeech.js"></script>
  <title>read aloud</title>

</head>
 <body>
  <header>

<a href="https://cdpn.io/chandani7021/debug/eYzLWPy/LDMmdndDRXXk"><img
src="https://cdn2.iconfinder.com/data/icons/multimedia-sound-7/48/Audio-_play-_sound-
_voice-512.png"  height="50" width="100" title="tts" class="corners" ></a>

  <br>
    <div class="topnav">
 <a href="https://cdpn.io/chandani7021/debug/eYzLWPy/LDMmdndDRXXk">HOME</a>
 <a
href="https://cdpn.io/chandani7021/debug/NWrVMdQ/mWMoNzNjBywk">SOFTWARE</a
>
<a href="http://127.0.0.1:8000/contact">CONTACT US</a>
      </div>
  </header>



   <h1> <em>Convert Your Texts <br>Into Voice</b></h1>
  <h2>Instantly read out your texts with natural sounding voices ONLINE!!!</h2>
  </div>

    <br>
      <div class="form">
        <br>
        <br>
        <br>
 <textarea name="styled-textarea" id="text" onfocus="this.value=''; setbg('#e5fff3');"
onblur="setbg('white')" rows="30" cols="100" title="ENTER TEXT" spellcheck="true">Enter
your text here...</textarea>
```

```
      <div class="slidecontainer">

<label for="voices">Voice:<br><select name="voices" id="voices" class="select-css" title="Selects TTS voice"></select></label>

    <br>
       <br>

          <label for="pitch" title="Sets the TTS pitch">Pitch: <span>1</span><br><input type="range" name="pitch" id="pitch" min="0" max="2" step="0.01" value="1" ></label><br>


   <label for="rate" title="Sets the TTS rate">Rate: <span>1</span><br><input type="range" name="rate" id="rate" min="0" max="10" step="0.01" value="1"></label><br>

 <label for="volume" title="Sets the TTS volume">Volume: <span>1</span><br><input type="range" name="volume" id="volume" min="0" max="1" step="0.01" value="1"></label><br><br>
  <br>
     <div id="wordoutput" title="Shows the word spoken if 'onboundary' is supported"></div>
      </div>
<br>
  <button id='talk' class="button" button type="button" title="LISTEN"><span>Speak it!! </span></button>
        <br>
    </div>
   </div>
  <br>

 <div class="abt">
  <section class="para">
    <br>
 <br><br>
    <br>
   <p><h2><i>MAIN FEATURES</i></h2><br>
This text to speech converter is a free text to speech converter that supports all modern browsers,<br> includeing chrome,firefox etc.<br>
         It includes multiple language accents.
    </p>
     <br>

     <h2>FUN, ONLINE,FREE</h2>
   <p> directly copy text and click the speak it!!! buton. that's it. No downloads. No logins. No fus. No password.<br>
     listen to great content with fun . Great for proof-reading. Great for kids and more.
    <br>

    <h2>NATURAL VOICES</h2><br>
    We have high-quality natural sounding voices.<br>
```

there are male and female voices in different accents.
Choose the voice you like,insert text and click speak it!!! to generate the synthesized speech and enjoy listening.</p></section>

</body>
</html>

CSS CODE:

```
body{
  margin: 1%;
  padding:1%;
background: rgb(36,58,94);
background: linear-gradient(90deg, rgba(36,58,94,1) 26%, rgba(51,90,136,1) 86%, rgba(13,48,90,1) 100%);
background-size:cover;
    background-repeat:no repeat;
   -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

.topnav{
  overflow:hidden;
   margin: 0;
  padding:0;
  border:0;
  font-family: "Comic Sans", Comic Sans MS,cursive;
background: #0f0c29;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #24243e, #302b63, #0f0c29);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-gradient(to right, #24243e, #302b63, #0f0c29); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */

  border-radius:10px;
    -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

.topnav a {
  float: left;
  color:  #f1f1f1;
  text-align: center;
  padding:20px 40px;
  font-size: 22px;
}

.topnav a:hover {
  background-color: #ddd;
  color: black;
```

```
}

h1{
  font-size:3.5rem;
  color:#ffffff;
  display: block;
  text-align: center;

}
h2{
  display:inline-block;
  color:#ffffff;
  text-align:center;
  }
 img.corners{
    border-radius: 25px;
  padding: 20px;
  float:left;
  }

.group{
  float:left;
  display:inline-block;
}

.form{
  display:inline-block;
 background: rgb(13,48,90);
background: linear-gradient(90deg, rgba(13,48,90,1) 23%, rgba(51,90,136,1) 71%,
rgba(13,48,90,1) 100%);
  height:30rem;
  width:100%;
  border: 2.5px solid #1c1b1c;
  border-radius:15px;
}
 textarea#text{
  width: 50vw;
  height: 200px;
  border: 3px solid #cccccc;
   position:absolute;
   left:50px;
 font-family: Tahoma, sans-serif;
   font-size:1rem;
background-color:#e5fff3
  background-repeat: no-repeat;
   display:inline-block;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}
```

```
.slidecontainer {
display: inline-block;
float:right;
            color: #ddd;
    -moz-appearance: none;
 -webkit-appearance: none;
  appearance: none;
}
input[type="range"] {
            -webkit-appearance: none;
            -webkit-tap-highlight-color: rgba(255, 255, 255, 0);
  display:inline-block;
            width: 350%;
            height: 10px;
            margin: 0;
            border: none;
            padding: 1px 2px;
            border-radius: 14px;
            background: #f2f7f4;
            box-shadow: inset 0 1px 0 0 #0d0e0f, inset 0 -1px 0 0 #3a3d42;
            -webkit-box-shadow: inset 0 1px 0 0 #0d0e0f, inset 0 -1px 0 0 #3a3d42;
            outline: none; /* no focus outline */
}

input[type="range"]::-moz-range-track {
            border: inherit;
            background: transparent;
}

input[type="range"]::-ms-track {
            border: inherit;
            color: transparent; /* don't drawn vertical reference line */
            background: transparent;
}

input[type="range"]::-ms-fill-lower,
input[type="range"]::-ms-fill-upper {
            background: transparent;
}

input[type="range"]::-ms-tooltip {
            display: none;
}

/* thumb */

input[type="range"]::-webkit-slider-thumb {
            -webkit-appearance: none;
             display:inline-block;
  width: 40px;
```

```
		height: 18px;
		border: none;
		border-radius: 12px;
		background-image: -webkit-gradient(
			linear,
			left top,
			left bottom,
			color-stop(0%, #529de1),
			color-stop(100%, #245e8f)
		); /* android <= 2.2 */
		background-image: -webkit-linear-gradient(
			top,
			#529de1 0,
			#245e8f 100%
		); /* older mobile safari and android > 2.2 */
		background-image: linear-gradient(
			to bottom,
			#529de1 0,
			#245e8f 100%
		); /* W3C */
}
input[type="range"]::-moz-range-thumb {
		width: 40px;
		height: 18px;
  display:inline-block;
		border: none;
		border-radius: 12px;
		background-image: linear-gradient(
			to bottom,
			#529de1 0,
			#245e8f 100%
		); /* W3C */
}

input[type="range"]::-ms-thumb {
		width: 40px;
		height: 18px;
  display:inline-block;
		border-radius: 12px;
		border: 0;
		background-image: linear-gradient(
			to bottom,
			#529de1 0,
			#245e8f 100%
		); /* W3C */
}


.select-css{
 display: inline-block;
 font-size: 17px;
```

```
  font-family: sans-serif;
  font-weight: 700;
  color: #0f0c29;
  line-height: 100%;
  padding: .6em 1.4em .5em .8em;
  width: 100%;
  max-width: 100%;
  float:right;
  box-sizing: border-box;
  margin: 0;
  border: 1px solid #aaa;
  box-shadow: 0 1px 0 1px rgba(0,0,0,.04);
  border-radius: .5em;
  -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
  background-color: #fff;
}
.select-css:-ms-expand {
  display: inline-block;
}
.select-css:hover {
  border-color: #fff;
}
.select-css:focus {
  border-color: #aaa;
  box-shadow: 0 0 1px 3px rgba(59, 153, 252, .7);
  box-shadow: 0 0 0 3px -moz-mac-focusring;
  color: #222;
  outline: none;
}
.select-css option {
  font-weight:normal;
}

label{
  display:inline-block;
  font-size:22px;
}

#wordoutput {
 height: 4rem;
  width:90%;
        border: 1px solid #858c87;
  border-radius:10px;
        text-align: center;
        font-size: 3rem;
        line-height: 3rem;
 }

  #talk {
```

```
  display: inline-block;
  border-radius: 10px;
 background-color: #e6494f;
  border: none;
  color: #FFFFFF;
  text-align: center;
  text-shadow: 0px -1px 0px #2b665e;
  font-size: 2.8vw;
  padding: 8px 10px;
  width: 18%;
    position:absolute;
    top:700px;
    right:750px;
  cursor: pointer;
   -moz-appearance: none;
 -webkit-appearance: none;
  appearance: none;
}


    .button span {
  cursor: pointer;
  display: inline-block;
  position: absolute;
  transition: 0.5s;
}

.button span:after {
  content: '\00bb';
  position: absolute;
  opacity: 0;
  top: 0;
  right: -20px;
  transition: 0.5s;
}

.button:hover span {
  padding-right: 25px;
}

.button:hover span:after {
  opacity: 1;
  right: 0;
}

.para{
  display:inline-block;
  position:absolute;
  top:750px;
  left:40px;
 font-size:22px;
```

```
 color:#ffffff;
}
```

JAVASCRIPT CODE:

```
(function() {
	if (!"speechSynthesis" in window) {
		Document.write("Sorry. Your browser does not have speech support");
		exit;
	}

	var synth = window.speechSynthesis;
	console.log("speechSynthesis", synth);

	// Test to see when the "onvoiceschanged" is triggered
	synth.onvoiceschanged = function(e) {
		console.log("onvoiceschanged triggered", e);
	};

	// Chrome loads voices asynchronously.
	// Chrome and Edge loads the voices asynchronously (But not Firefox), so we have to wait for them to appear
	// Another way would be to use the "synth.onvoiceschanged" callback function.
	// But this have some drawbacks (In Chrome the event is triggered everytime you use a Google voice), so Im using the timer method.
	var waitTimerID = setInterval(function() {
		console.log("ZZzz..");

		// lang, default, name, voiceURI, localService
		var voices = synth.getVoices();

		// We got voices!
		if (voices.length != 0) {
			// Clear the interval timer, as we have the data needed
			clearInterval(waitTimerID);

			// Sort voices by language, then name (For Selectbox)
			voices.sort(function(obj1, obj2) {
				if (obj1.lang < obj2.lang) return -1;
				if (obj1.lang > obj2.lang) return 1;
				if (obj1.name < obj2.name) return -1;
				if (obj1.name > obj2.name) return 1;
				return 0;
			});

			var textInput = document.querySelector("#text");
			var wordOutput = document.querySelector("#wordoutput");


			var voicesSelect = document.querySelector("#voices");
```

```
var pitch = document.querySelector("#pitch");
var rate = document.querySelector("#rate");
var volume = document.querySelector("#volume");
var button = document.querySelector("#talk");

// Test if local storage is available
var lStorage = null;
if (
	!"localStorage" in window ||
	typeof window.localStorage !== "undefined"
) {
	// Accessing "window.localStorage" on file: protocol, throws a "SCRIPT16389: Unspecified error." in MS Edge.
	// So it needs the extra "typeof" check.
	lStorage = window.localStorage;
} else {
	console.log("No localstorage support available!");
}

// Get word at specific position. Used for extracting the word currently spoken
// Source: https://stackoverflow.com/questions/5173316/finding-the-word-at-a-position-in-javascript
var getWordAt = function(str, pos) {
	// Perform type conversions.
	str = String(str);
	pos = Number(pos) >>> 0;

	// Search for the word's beginning and end.
	var left = str.slice(0, pos + 1).search(/\S+$/),
		right = str.slice(pos).search(/\s/);

	// The last word in the string is a special case.
	// else Return the word, using the located bounds to extract it from the string.
	return right < 0 ? str.slice(left) : str.slice(left, right + pos);
};

// OnInput callback handler for range input controls
var onInput = function(e) {

	e.target.previousElementSibling.previousElementSibling.innerHTML =
		e.target.value;
	saveToLocalStorage();
};

// OnChange callback handler for voice selectbox
var onSelect = function(e) {
	var curVoice = voices.find(
		x => x.name === voicesSelect[voicesSelect.selectedIndex].value
```

```
                    );

                    // Reduce max rate if not a localService
                    var maxRate = curVoice.localService ? 10 : 2;
                    rate.setAttribute("max", maxRate);

                    var reset = 1;
                    pitch.value = reset;

        pitch.previousElementSibling.previousElementSibling.innerHTML = reset;

                    rate.value = reset;

        rate.previousElementSibling.previousElementSibling.innerHTML = reset;

                    volume.value = reset;

        volume.previousElementSibling.previousElementSibling.innerHTML = reset;

                    saveToLocalStorage();

                    setVoiceInfo();
                };

                // OnBoundary callback handler for when words is spoken
                // Note: Event is not triggered when "localService" is false (Like the
Google voices in Chrome)
                var onBoundary = function(e) {
                    if (e.name == "word") {
                        var word = getWordAt(e.target.text, e.charIndex);
                        wordOutput.innerHTML = word;
                    }
                };

                // OnEnd callback handler for when speaking ends
                var onEnd = function(e) {
                    wordOutput.innerHTML = "";
                };

                // Trigger the SpeechSynthesisUtterance
                var utterText = function() {
                    var curVoice = voices.find(
                        x => x.name ===
voicesSelect[voicesSelect.selectedIndex].value
                    );
                    // Cancel speaking if active
                    if (synth.speaking) synth.cancel();

                    var utterance = new SpeechSynthesisUtterance();
                    utterance.voice = curVoice;
                    utterance.lang = curVoice.lang;
```

```
                utterance.pitch = pitch.value;
                utterance.rate = rate.value;
                utterance.volume = volume.value;
                utterance.text = textInput.value;
                if (curVoice.localService) utterance.onboundary =
onBoundary;
                else {
                        wordOutput.innerHTML = "😄";
                }
                utterance.onend = onEnd;
                synth.speak(utterance);
        };

        var saveToLocalStorage = function() {
                if (lStorage == null) return;
                lStorage.setItem("voice",
voicesSelect[voicesSelect.selectedIndex].value);
                lStorage.setItem("pitch", pitch.value);
                lStorage.setItem("rate", rate.value);
                lStorage.setItem("volume", volume.value);
        };

        /* Add events to input controls (range sliders) and voice select */
        volume.addEventListener("input", onInput);
        pitch.addEventListener("input", onInput);
        rate.addEventListener("input", onInput);
        voicesSelect.addEventListener("change", onSelect);

        button.addEventListener("click", utterText);

        // Set initial position of range input sliders
        var pVal = (rVal = vVal = 1);
        if (lStorage !== null) {
                pVal = lStorage.getItem("pitch") || 1;
                rVal = lStorage.getItem("rate") || 1;
                vVal = lStorage.getItem("volume") || 1;
        }
        pitch.value = pVal;
        pitch.previousElementSibling.previousElementSibling.innerHTML =
pVal;

        rate.value = rVal;
        rate.previousElementSibling.previousElementSibling.innerHTML =
rVal;

        volume.value = vVal;

    volume.previousElementSibling.previousElementSibling.innerHTML = vVal;

        // Remove existing items from selectbox
        for (var i = voicesSelect.options.length - 1; i >= 0; i--) {
```

```
                        voicesSelect.remove(i);
                    }

                    // Populate selectbox
                    var voVal = lStorage !== null ? lStorage.getItem("voice") : "";
                    for (var i = 0; i < voices.length; i++) {
                        var option = document.createElement("option");
                        option.textContent = voices[i].name + " (" + voices[i].lang +
")";
                        option.value = voices[i].name;
                        voicesSelect.appendChild(option);
                        if (voVal == voices[i].name) voicesSelect.selectedIndex = i;
                    }
                    setVoiceInfo();

                    // Reduce max rate if voice is not a localService
                    var curVoice = voices.find(
                        x => x.name ===
voicesSelect[voicesSelect.selectedIndex].value
                    );
                    var maxRate = curVoice.localService ? 10 : 2;
                    rate.setAttribute("max", maxRate);
                }
            }, 10);
})();
```

2.Software Page:-

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>

   <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" type="text/css" href="C:\Users\CHZ\Desktop\study\textspeech.css">
    <script src="C:\Users\CHZ\Desktop\study\\textspeech.js"></script>

      <title>read aloud</title>

</head>
  <body>
    <header>

 <div class="speech-img">
<div class="container text-center">
<a              href="https://cdpn.io/chandani7021/debug/eYzLWPy/LDMmdndDRXXk"><img
src="https://cdn2.iconfinder.com/data/icons/multimedia-sound-7/48/Audio-_play-_sound-
_voice-512.png"  height="50" width="100" title="tts" class="corners" ></a>

  <div class="topnav">
        <a
href="https://cdpn.io/chandani7021/debug/eYzLWPy/LDMmdndDRXXk">HOME</a>
  <a
href="https://cdpn.io/chandani7021/debug/NWrVMdQ/mWMoNzNjBywk">SOFTWARE</
a>
<a href="http://127.0.0.1:8000/contact">CONTACT US</a>
     </div>


  <img  src="https://www.naturalreaders.com/comfiles/software/img/test.svg"      alt="apeak"
class="image" >
  <br>
  <br>
  <br>
 <div class="box">
  <h1>At a Glance</h1>
  <ul>
    <li>Text-to-speech technology reads digital text loudly from computer,smartphones and
tablets.</li><br>
    <li>this technology can help kids who struggle with reading.</li>
  </ul>
  </div>
  <br>
  <br>
  <p>Text-to-speech Converter is a type of <b><em>assistive technology</em></b> that reads
digital texts aloud.It is also called <b>"Read aloud Technology".</b>With a click on Speak
button it will take word on a computer or other digital device and convert it into audio.<br>
```

```
   this converter is very helpful for kids who struggle with reading but it can also help kids with focusing.</b>

   <h2>How It Works</h2>
   <p>This text to Speech converter works with nearly every personal digital device including computers,smartphones and tablets.<br>
    The voice in converter is natural human like. The voice qualities varies.</p>
   <br>
   <h2>How Text To Speech converter can help your Child</h2>
   <p>
   <ul><li>Improves word recognition</li>
   <li>Increases the ability to pay attention and remember information while reading</li>
   <li>Allows kids to focus on comprehension instead of sounding out words</li>
   <li>Increases kids staying power for reading assignments</li>
   <li>Helps kids recognize and fix errors in their own writing</li>
   <br>
   <br>
   <h4><b>"LIKE AUDIO BOOKS THIS CONVERTER WON'T SLOW DOWN THE DEVELOPMENT OF YOUR CHILD'S READING SKILLS. AND ALSO WILL IMPROVE LISTENING SKILLS"</b></h4>
   </p>
   </body>
</html>
```

CSS CODE:-

```
body{
  margin: 1%;
  padding:1%;
background: rgb(36,58,94);
background: linear-gradient(90deg, rgba(36,58,94,1) 26%, rgba(51,90,136,1) 86%, rgba(13,48,90,1) 100%);
background-size:cover;
    background-repeat:no repeat;
   -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

.topnav{
  overflow:hidden;
   margin: 0;
  font-family: "Comic Sans", Comic Sans MS,cursive;
background: #0f0c29;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #24243e, #302b63, #0f0c29);  /* Chrome 10-25, Safari 5.1-6 */
```

```
background: linear-gradient(to right, #24243e, #302b63, #0f0c29); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */

  border-radius:10px;
    -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

.topnav a {
  float: left;
  color:  #f1f1f1;
  text-align: center;
  padding:20px 40px;
  font-size: 22px;
}

.topnav a:hover {
  background-color: #ddd;
  color: black;
}

.sign{
 position:absolute;
  right:30px;
  border-radius:5px;
  background:background: #ee0979;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #ff6a00, #ee0979);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-gradient(to right, #ff6a00, #ee0979); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */;
}

h1{
  font-size:3.5rem;
  color:#ffffff;
  display: block;
  text-align: center;

}
h2{
  display:inline-block;
  color:#ffffff;
  text-align:center;
  }
 img.corners{
```

```
    border-radius: 25px;
  padding: 20px;
  float:left;
  }
.image{
  padding: 20px;
  float:right;
  height:30vw;
  width:30vw;
  -moz-appearance: none;
 -webkit-appearance: none;
  appearance: none;
}

.box{
  display:inline=block;
   width: 50%;
  padding: 20px;

  background:#2b2f33;
  border: 5px solid gray;
  margin: 0;
  border-radius: 20px;
   -moz-appearance: none;
 -webkit-appearance: none;
  appearance: none;

}

ul{
  display:inline-block;
  color:#fcfcff;
  font-size:22px;
    -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

p{
  display:block;
  color:#fcfcff;
  font-size:22px;
  padding:0.5px 1px ;
    -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
```

```
}
h4{
    display:block;
  color:#fcfcff;
  font-family:"Comic Sans", Comic Sans MS,cursive;
  font-size:22px;
  padding:0.5px 1px ;
    -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

/* For mobile phones: */
[class*="col-"] {
  width: 100%;
}
@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

3.Contact Page:-

Contact form in php:

```
<?php

namespace App\Http\Livewire;
```

```
use App\Models\Contact;
use Livewire\Component;

class ContactForm extends Component
{
   public $name = '';
   public $emailid = '';
   public $phone = '';
   public $website = '';
   public $message = '';

   public function postContact()
   {
      Contact::create([
         'name' => $this->name,
         'emailid' => $this->emailid,
         'phone' => $this->phone,
         'website' => $this->website,
         'message' => $this->message
      ]);

      $this->name = '';
      $this->emailid = '';
      $this->phone = '';
      $this->website = '';
      $this->message = '';
   }

   public function render()
   {
      return view('livewire.contact-form');
   }
}
```

Contact code in php:-

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Contact extends Model
```

```
{
    use HasFactory;

    protected $fillable = ['name', 'emailid', 'phone', 'website', 'message'];
}
```

Creating Contact table in php:-

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateContactsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('contacts', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('emailid');
            $table->string('phone');
            $table->string('website');
            $table->text('message');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('contacts');
    }
}
```

Text to speech module in php:-

```
<?php

namespace App\Http\Livewire;

use Livewire\Component;

class TextToSpeechModule extends Component
{
   public function render()
   {
      return view('livewire.text-to-speech-module');
   }
}
```

.env file:-

```
APP_NAME="Text To Speech"
APP_ENV=local
APP_KEY=base64:MVPkikMYSXOko47btwSmLUyDkEWyz94JiBaoKLOyC3Y=
APP_DEBUG=true
APP_URL=http://text-to-speech-no-login.test

LOG_CHANNEL=stack
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=text_to_speech
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
```

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

5.1 Working Of the System

Firstly We made a html file and styled it using CSS . In which We have added following things:-

1.Text Field : Text fields let users enter and edit text. Text fields allow users to enter text into a UI. They typically appear in forms and dialogs. A text field is an input field that merchants can type into. It has a range of options and supports several text formats including numbers.

2. Pitch: In music, position of a single sound in the complete range of sound. ... Sounds are higher or lower in pitch according to the frequency of vibration of the sound waves producing them. Pitch is a perceptual property of sounds that allows their ordering on a frequency-

related scale, or more commonly, pitch is the quality that makes it possible to judge sounds as "higher" and "lower" in the sense associated with musical melodiess.
3. Rate: In audio production, a sample rate (or "sampling rate") defines how many times per second a sound is sampled. Technically speaking, it is the frequency of samples used in a digital recording. The standard sample rate used for audio CDs is 44.1 kilohertz.
4. Volume: The volume of a sound is how loud or quiet the sound is. Sounds are vibrations that travel through the air. A nail hit hard with a hammer will make a strong vibration, which means it will make a loud sound.
5. select a voice: For user to sleect the particular voice in which he wanted to listen their inputted text.

For our project to give output as a voice from inputted text we made a function in JAVASCRIPT and check if the browser user used does support speech synthesis if does then the synthesis will work properly if not then a It shows that “Sorry. Your browser does not have speech support”.
After checking speech support we added speechSynthesis(); by declaring variable as synth.
We have used onvoicechanged(); to change the voice of inputted text.
We have added Voice Information with the help of voiceInfo();.
We have declared some variables and linked them with textarea, voices, pitch, rate, volume and button.
We have used document.querySelecter(); to get input from textarea and wordoutput.
We used addEventListener(); to take input from user rate,pitch and volume and added setVoiceInfo(); to set the information of voice.
We used SpeechSynthesisisUtterance(); to get inputted text into speech as output.
And here Our whole process of conversion of text to speech is completed.

For contact page we have used Laravel application. Connected it to database using .env file(environment file) we added names in .env file .
Then we run XAMPP added component-contact, text to speech and then made contact models to save in database. These models provide structure. When we submit comment It requests a server and store the submitted information in the server.

Chapter 6

6.1 Methodology

text to Speech conversion system Methodology:-
1. Speech synthesis techniques is used in order to get the naturalness quality in the synthetic speech.
2. The process of the English language can be Used as the basic unit for speech synthesis.
3. Speech database for the English language will be developed using phoneme.
4. The input text will be separated into English Phoneme.
5. Phonemes will be searched in the database and corresponding phonemes sounds will be Concatenated to generate synthesized output Speech.

6.2 Result of Experiment:-

The inputted text first generates into voice form after text analysis and phoneme analysis and then user gets to hear a voice as an output.
We successfully implemented the JAVASCRIPT CODE and got the output of Inputted text as a voice.

Chapter 7

7.1 Project Screenshots

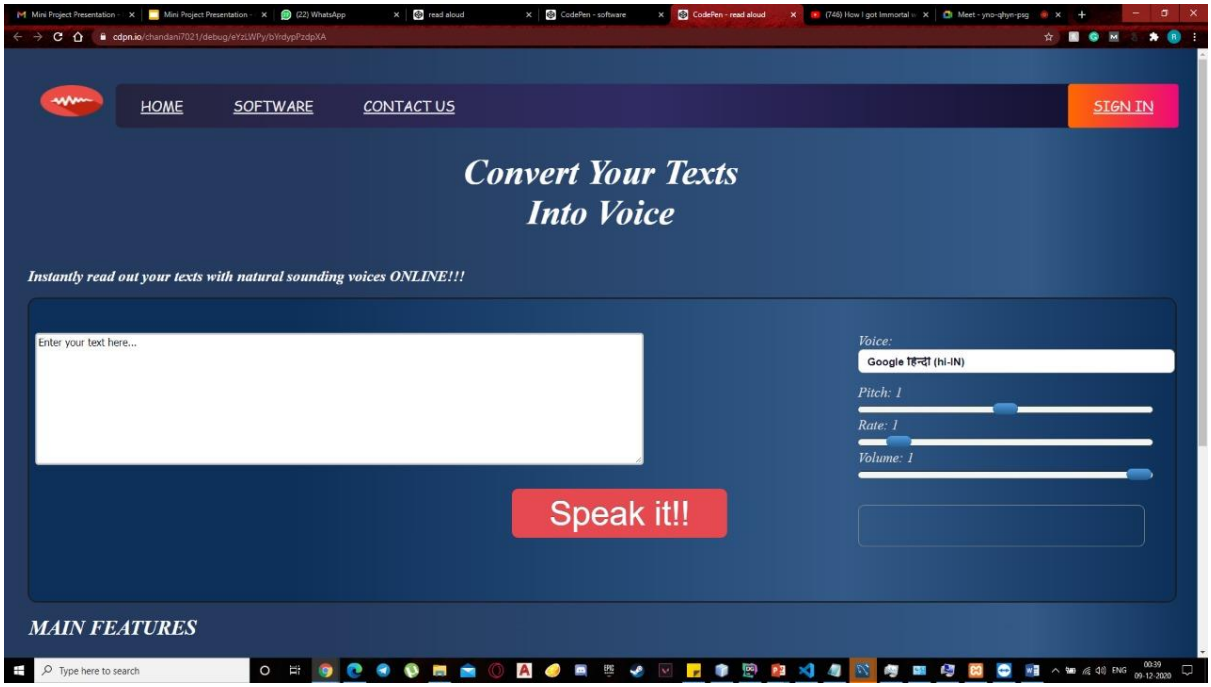


Fig 7.1 Home Page

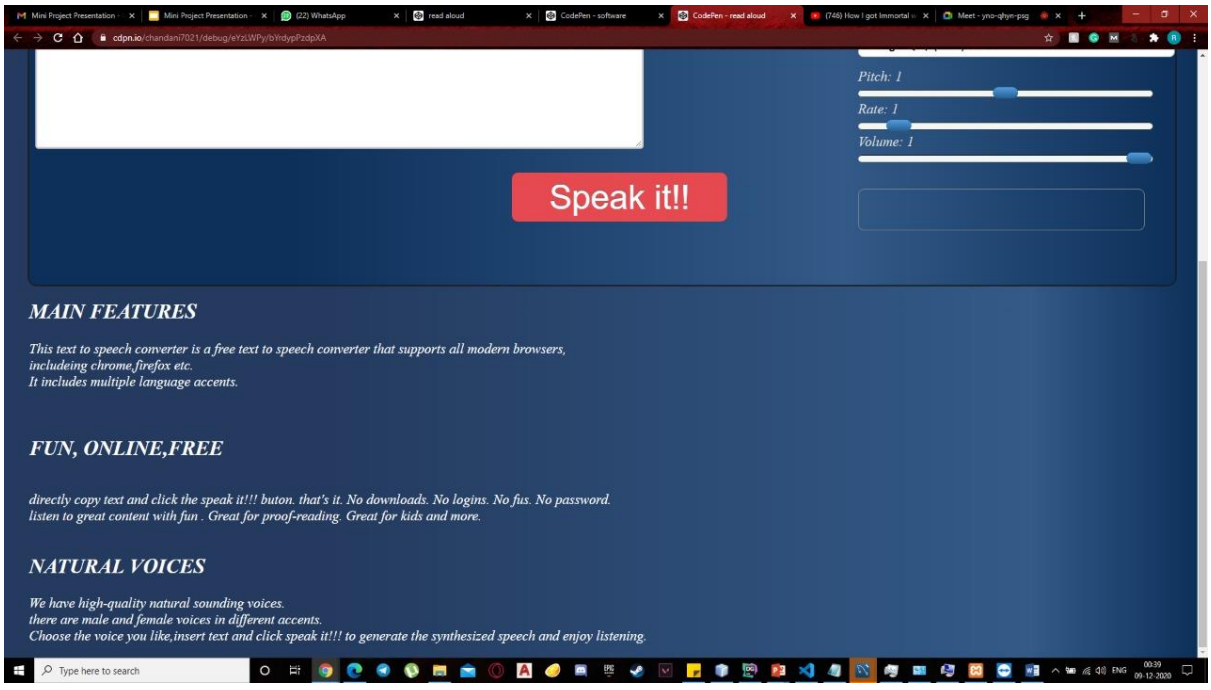


Fig 7.2 Home Page

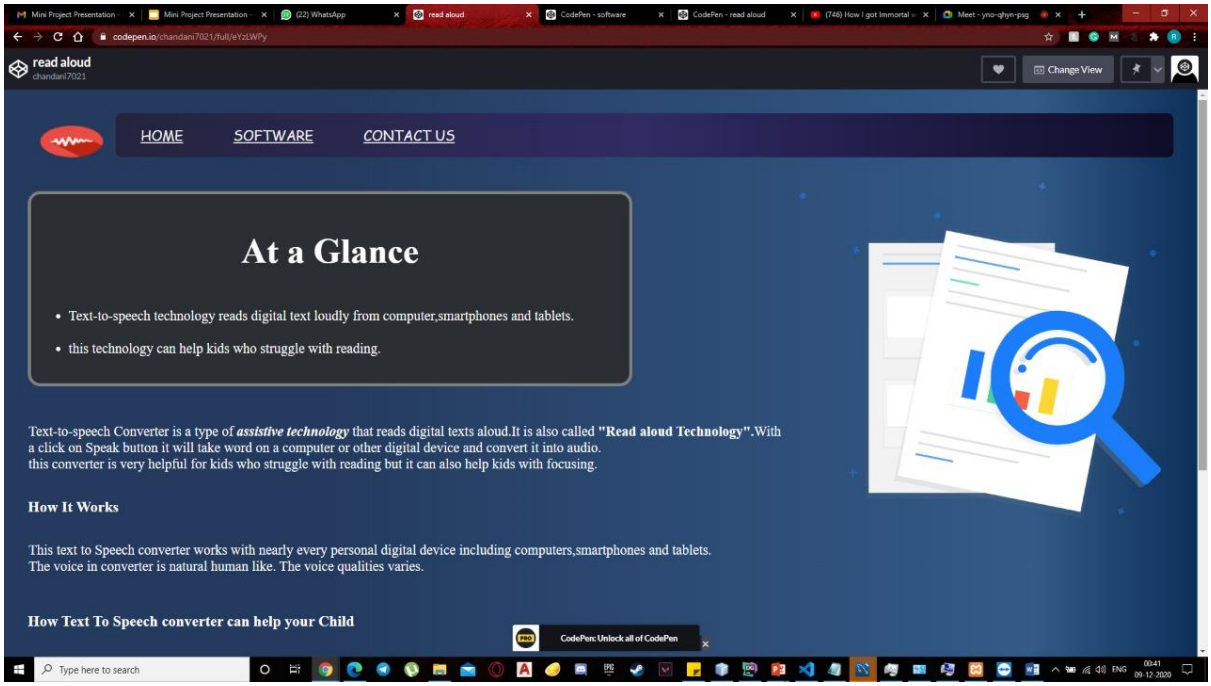


Fig 7.3 Software Page

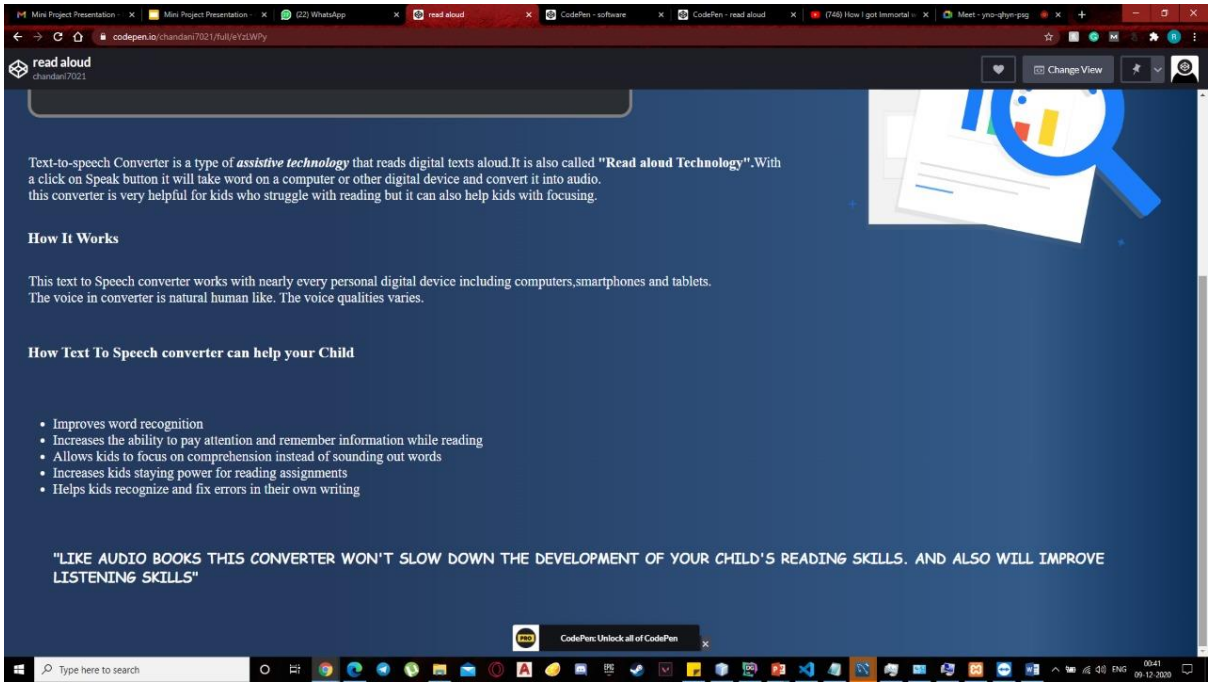


Fig. 7.4 Software Page

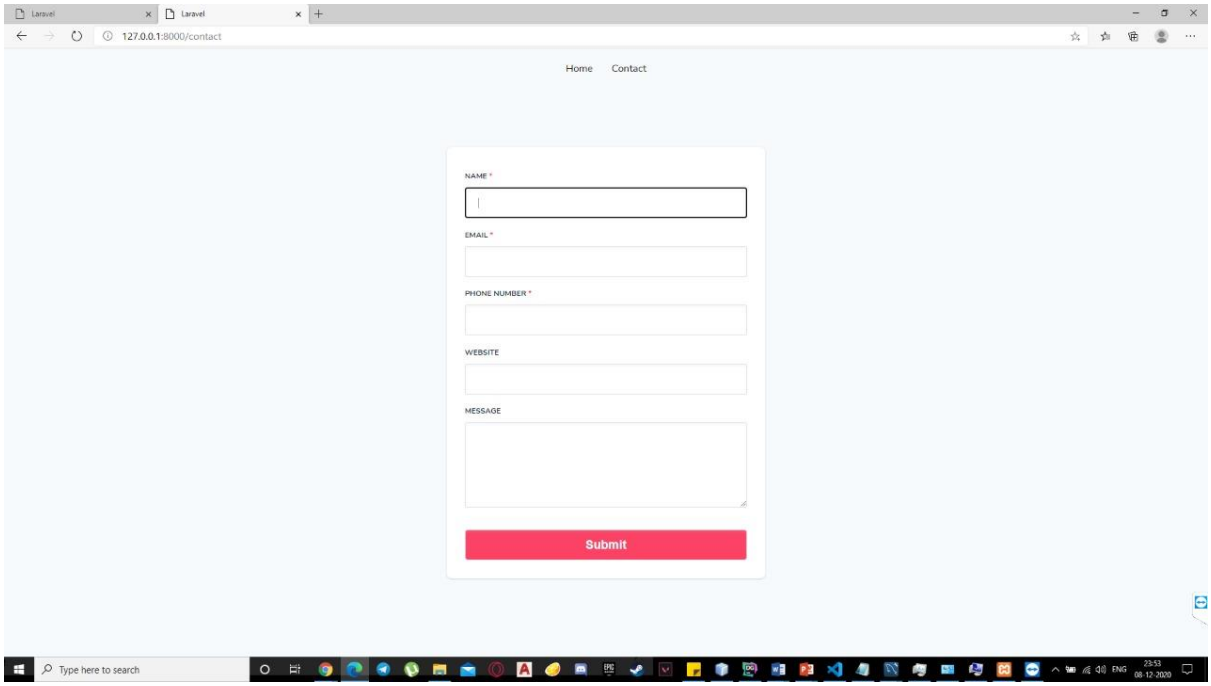


Fig 7.5 Contact Page

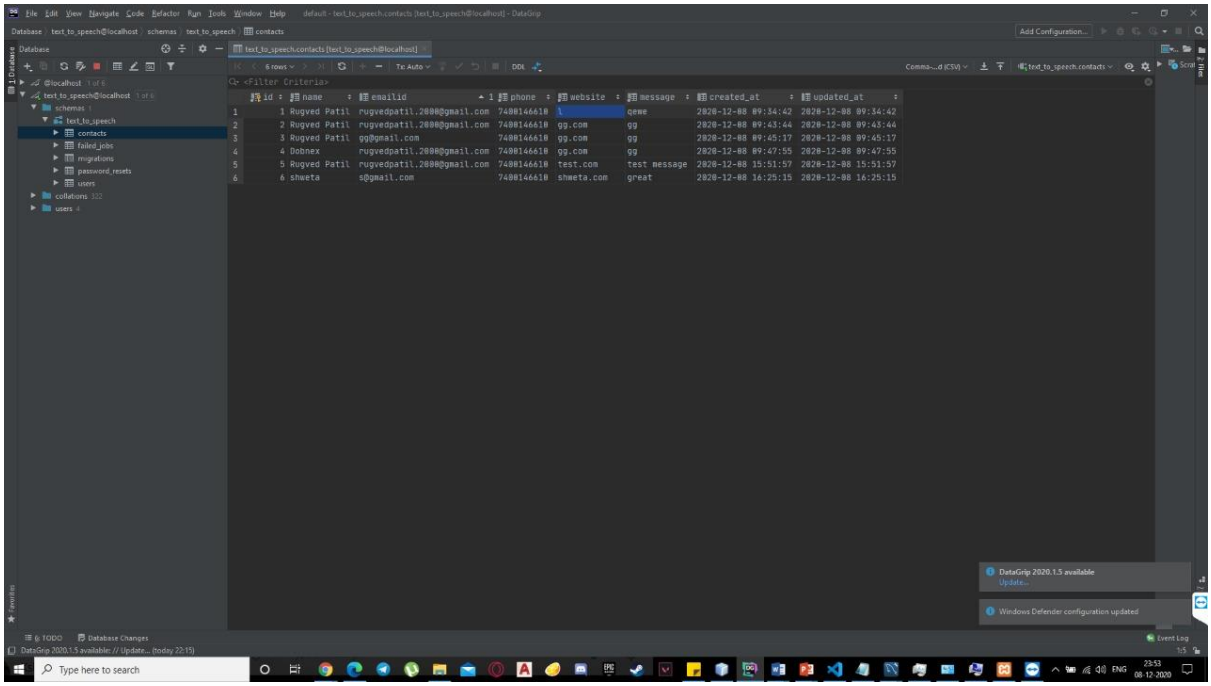


Fig 7.6 Contact Database Page

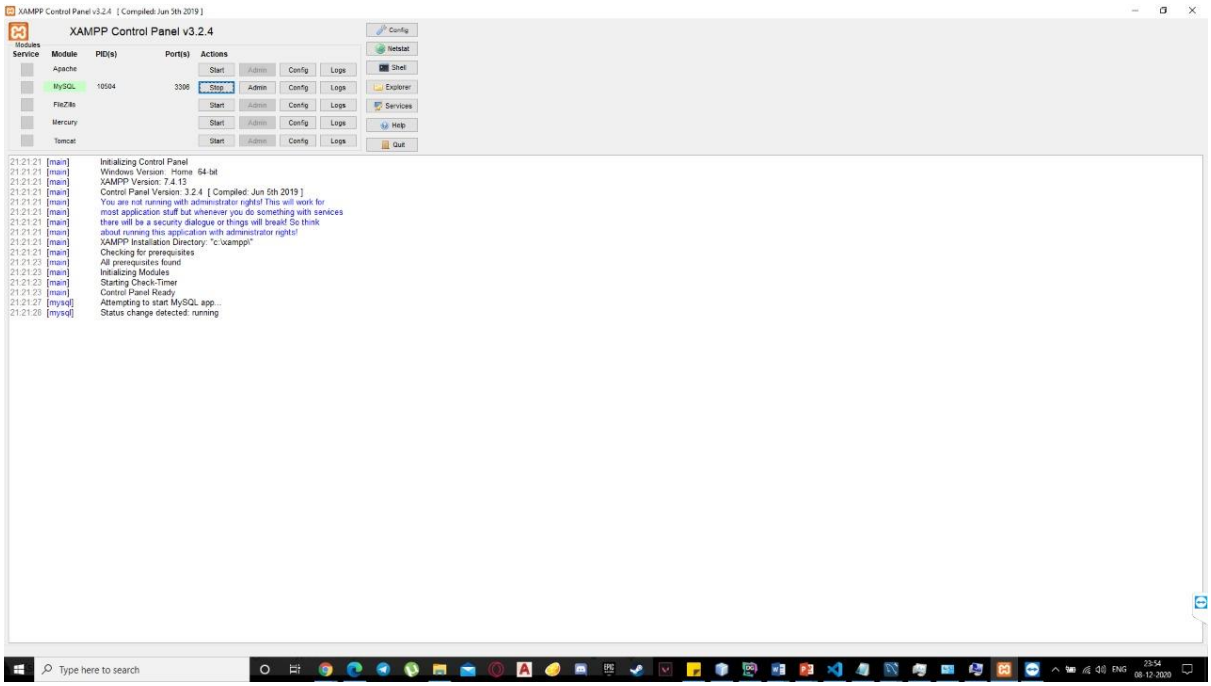


Fig. 7.7 XAMPP Control Panel

Chapter 8

Conclusion

Text to speech synthesis is a rapidly growing aspect of computer technology and is increasingly playing a more important role in the way we interact with the system and interfaces across a variety of platforms. We have identified the various operations and processes involved in text to speech synthesis. We have also developed a very simple and attractive graphical user interface which allows the user to type in his/her text provided in the text field in the software. Our system interfaces with a text to speech engine developed for English. In future, we plan to make efforts to create engines for localized Hindi and Marathi language so as to make text to speech technology more accessible to a wider range in Maharashtra. Another area of further work is the implementation of a text to speech system on other platforms, such as telephony systems, ATM machines, video games and any other platforms where text to speech technology would be an added advantage and increase functionality.

Reference:

1.We learned what is text to speech technology and how it works from:
https://www.understood.org/en/school-learning/assistive-technology/assistive-technologies-basics/text-to-speech-technology-what-it-is-and-how-it-works

2.We got information about speech synthesis from:
https://en.wikipedia.org/wiki/Speech_synthesis

3.Information about how does speech synthesis works in quiet easy manner given by:
https://www.explainthatstuff.com/how-speech-synthesis-works.html