1. INTRODUCTION

1.1 ABOUT THE PROJECT

A Text-to-speech synthesizer is an application that converts text into spoken word, by analyzing and processing the text using Natural Language Processing (NLP) and then using Digital Signal Processing (DSP) technology to convert this processed text into synthesized speech representation of the text. Here, we developed a useful text-to-speech synthesizer in the form of a simple application 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.

1.2 OVERVIEW OF THE PROJECT

Speech synthesis can be described as 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. The quality of a speech synthesizer is judged by its similarity to human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer.

2. SYSTEM SPECIFICATION

2.1 HARDWARE CONFIGURATION

This section gives the details and specification of the hardware on which the system expected to work.

Processor	:	12^{th} GEN Intel(R) Core i3
Memory	:	16.0 GB RAM
Monitor Resolution	:	15.6 inch HD Monitor
Keyboard	:	100 keys
Mouse	:	Scroll/Optic with three buttons

2.2 SOFTWARE CONFIGURATION

This section gives the details of the software that are used for the development.

Front-End	:	HTML, CSS, and JavaScript
Operating system	:	Windows 11

2.3 FEATURES OF PROGRAMMING TOOLS

ABOUT HTML

HTML stands for Hyper Text Markup Language. It is the standard markup language used to create web pages. HTML describes the structure of a web page and consists of a series of elements that label pieces of content such as headings, paragraphs, links.

HTML is used to create the basic structure of a web page and is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

FEATURES OF HTML

Simple and user-friendly: HTML uses tags to structure and format the content of a web page. Tags are annotations that start with < and end with >.

Semantic structure: HTML5 introduced many new tags that have specific meanings and uses. For example, the <article> tag is used to mark a section of content that can stand on its own, such as a blog post or a news article.

Multimedia support: HTML can embed various types of media, such as images, videos, audio, animations, and more.

Interactivity: HTML can interact with other languages and technologies, such as CSS, JavaScript, PHP, AJAX, and more.

ABOUT CSS

CSS stands for Cascading Style Sheets. It is a style sheet language used to describe the presentation of a document written in HTML or XML 123. CSS is used to style and layout web pages, alter the font, color, size, and spacing of content, split it into multiple columns, or add animations and other decorative features.

FEATURES OF CSS

Opportunity in Web designing: If anyone wants to begin a career in web designing professionally, it is essential to have knowledge of CSS and HTML.

Website Design: With the use of CSS, we can control various styles, such as the text color, the font style, the spacing among paragraphs, column size and layout, background color and images, design of the layout, display variations for distinct screens and device sizes, and many other effects as well.

Web Control: CSS has controlling power on the documents of HTML, so it is easy to learn. It is integrated with the HTML and the XHTML markup languages.

Compatibility: The cascading style sheet permits content to be upgraded for one or more device types.

ABOUT JAVASCRIPT

JavaScript is a high-level, interpreted programming language that is widely used for creating interactive web pages and applications. It was first introduced in 1995 by Brendan Eich, a programmer at Netscape Communications Corporation. JavaScript is a client-side scripting language, which means that it runs on the user's web browser rather than on the server. It is used to add interactivity to web pages, such as form validation, dynamic effects, and animations. JavaScript is also used for server-side programming through platforms such as Node.js.

FEATURES OF JAVASCRIPT

- Object-Centered Script Language
- Client Edge Technology
- Validation of User's Input
- Else and If Statement
- Interpreter Centered
- Ability to perform In Built Function
- Case Sensitive format
- Light Weight and delicate Statements
- Looping Handling Events
- Object-Centered Script Language
- Client Edge Technology
- Validation of User's Input
- Else and If Statement

3. SYSTEM STUDY

3.1 EXISTING SYSTEM

A system study for a text-to-speech (TTS) system using a frontend typically involves several key components and considerations:

User Requirements: Understand the specific needs of users for the TTS system. Determine the target audience, use cases, and the languages or accents the system should support.

Text Preprocessing: Analyze the input text data. Preprocess the text to handle punctuation, special characters, and formatting, if necessary.

Linguistic Analysis: Incorporate linguistic analysis tools to handle grammar, pronunciation, and prosody. This may include part-of-speech tagging and syntax analysis.

Phonetic Transcription: Develop a phonetic transcription system to convert text into phonemes. This is crucial for accurate pronunciation.

Voice Selection: Choose or create voices for the TTS system. Consider factors such as voice quality, gender, and accent. Voices are often generated using deep learning models

3.2 DRAWBACKS OF EXISTING SYSTEM

Robotic and Unnatural Voice: Many older TTS systems produce robotic and unnatural-sounding voices, which can be unpleasant to listen to and lack the natural cadence of human speech.

Limited Expressiveness: Older TTS systems may struggle to convey emotion or nuance in speech, making it challenging to generate engaging and context-appropriate audio.

Pronunciation Errors: They may have difficulty with proper pronunciation of words, especially for names, technical terms, or regional accents, leading to mispronunciations..

3.2 PROPOSED SYSTEM

A proposed system for text-to-speech (TTS) using a frontend involves several key components and considerations. Here's an outline for a hypothetical system:

1. User Interface (Frontend): Create an intuitive and user-friendly frontend, which could be a web application, mobile app, or desktop software. Provide an input area where users can type or paste the text they want to convert to speech.

2. Text Processing: Implement text preprocessing to clean and format the input text, handling punctuation and special characters.

3. Language and Voice Selection: Allow users to select from a range of available languages. Offer a variety of voices within each language, including different genders and accents.

4. Phonetic Transcription: Develop a phonetic transcription system to convert text into phonemes, ensuring accurate pronunciation.

4. SYSTEM DESIGN

4.1 INPUT DESIGN

Tokenization: Break the input text into individual words or units (tokens).

Normalization: Handle issues like capitalization, punctuation, and numerical representations.

Text cleaning: Remove unnecessary characters or symbols.

Part-of-speech tagging: Identify the grammatical category of each word.

Syntactic analysis: Parse the text to understand sentence structure.

Phonetic Transcription: Convert text to a phonetic representation. This helps with pronunciation.

4.2 OUTPUT DESIGN

Output Audio: Finally, produce the synthesized speech as an audio file in a suitable format (e.g., WAV or MP3) or as a real-time audio stream.

Voice Selection: Apply voice characteristics, including pitch, tone, and speaking style, based on the selected voice from the frontend.

Text-to-Speech Model: Use a neural network-based TTS model. Popular choices include Taciturn, Wave Net, or more recent models like Fast Speech and Parallel WaveGAN

5. CONCLUSION

In conclusion, text-to-speech (TTS) technology driven by advanced frontend systems has made significant advancements in recent years. These systems use cutting-edge natural language processing and deep learning techniques to convert text into high-quality, natural-sounding speech. The frontend components, including text preprocessing, linguistic analysis, and prosody modeling, play a crucial role in enhancing the overall TTS performance. As technology continues to evolve, we can expect even more realistic and versatile TTS systems that have far-reaching applications, from accessibility and assistive technology to voice assistants and content creation.

.

6. BIBLIOGRAPHY

- Taylor, P., & Black, A. W. (2017). Tacotron: Towards end-to-end speech synthesis. ArXiv preprint arXiv: 1703.10135.
- Zen, H., Senior, A., Schuster, M., Riedmiller, M., & Kain, A. (2019). Fast speech generation for TTS. ArXiv preprint arXiv: 1711.05747.
- Renals, S., Black, A. W., & Woodland, P. C. (2003). The HTK book. Cambridge University Engineering Department.

7. APPENDIX

A.CODING

Index1.html

```
<!DOCTYPE html>

<html>
   <head>
      <meta name="viewport" content="width=device-width,initial-scale=1.0">
      <title>Text To Speech Converter</title>
      <link rel="stylesheet" href="style1.css">
   </head>
   <body>
      <div class ="file">
         <h1>Text To Speech<span> Converter</span></h1>
         <textarea placeholder="TYPE HERE...."></textarea>
         <div class="row">
            <select></select>
            <button><img scr="images/play.png"> PLAY </button>
         </div>
      </div>
      <script src="script1.js"></script>
   </body>
</html>
```

style1.css

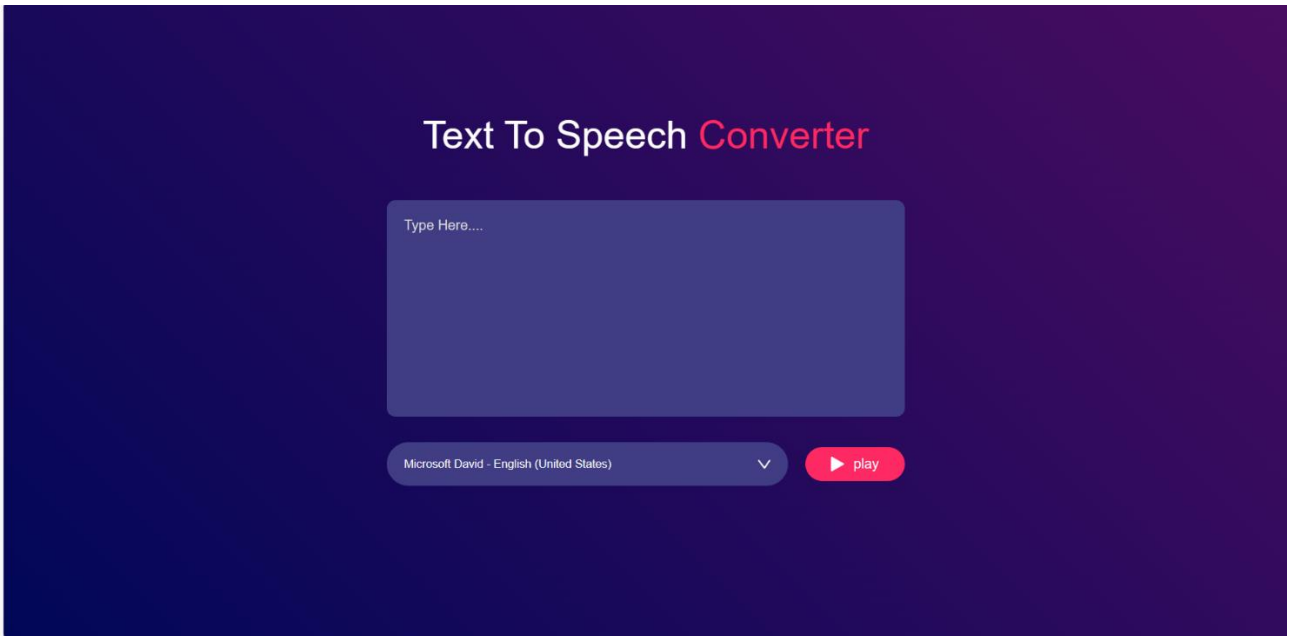
```
*{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        font-family: 'Poppins', sans-serif;
  }
  .file{
        width: 100%;
        min-height: 100vh;
        background: linear-gradient(45deg, #010758, #490d61);
        display: flex;
        align-items: center;
        justify-content: center;
        flex-direction: column;
        color: #fff;
  }
  .file h1{
        font-size: 45px;
        font-weight: 500;
        margin-top: -50px;
        margin-bottom: 50px;
  }
  .file h1 span{
        color: #ff2963;
  }
  textarea{
        outline: 0;
        width: 600px;
        height: 250px;
        background: #403d84;
        color: #fff;
        font-size: 15px;
        border: 0;
        border-radius: 10px;
        padding: 20px;
        resize: none;
        margin-bottom: 30px;
  }
 textarea::placeholder
```

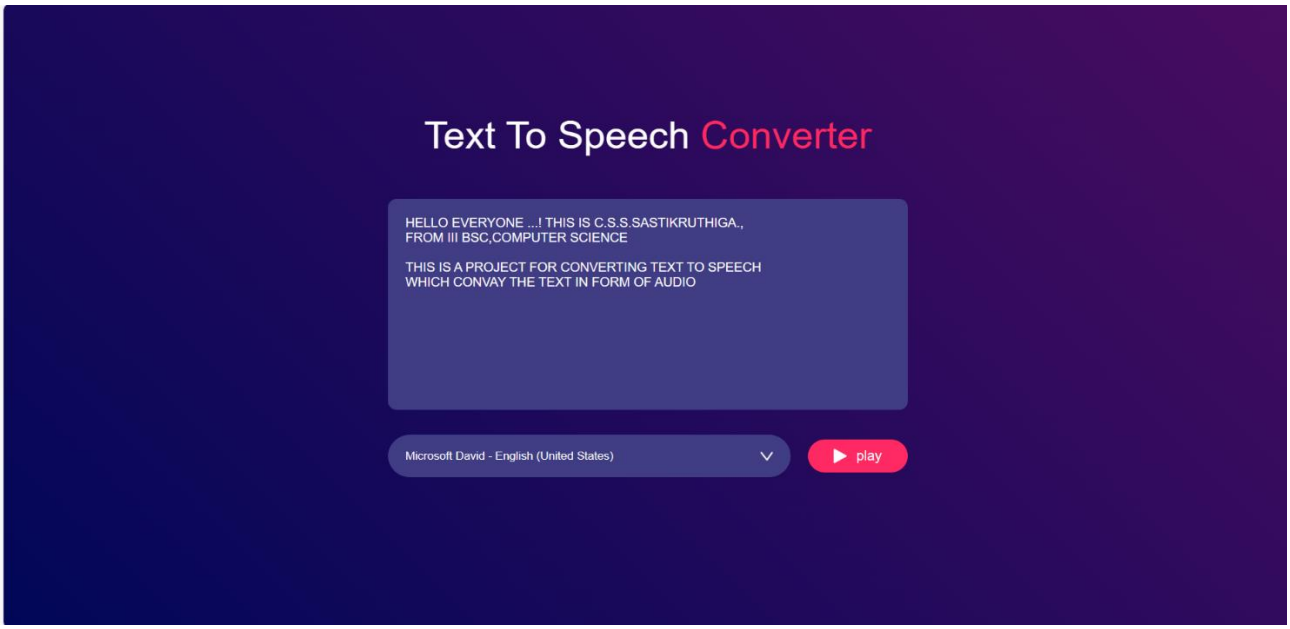
```
 {
        font-size: 16px;
        color: #ddd;
 }
.row{
        width: 600px;
        display: flex;
        align-items: center;
        gap: 20px;
  }
button{
        padding: 10px 30px;
        border-radius: 35px;
        color: #fff;
        font-size: 16px;
        cursor: pointer;
        border: 0;
        outline: 0;
        display: flex;
        align-items: center;
        background: #ff2963;
  }
button img{
        width: 16px;
        margin-right: 10px;
  }
select{
        flex: 1;
        color: #fff;
        background: #403d84;
        height: 50px;
        padding: 0 20px;
        outline: 0;
        border: 0;
        border-radius: 35px;
        appearance: none;
        background-image: url(images/dropdown.png);
        background-repeat: no-repeat;
        background-size: 15px;
        background-position-x: calc(100% - 20px);
        background-position-y: 20px;
  }
```

script1.js

```
let speech = new SpeechSynthesisUtterance();

let voices = [];

let voiceselect = document.querySelector("select");

window.speechSynthesis.onvoiceschanged = () => {
    voices = window.speechSynthesis.getVoices();
    speech.voice = voices[0];

    voices.forEach((voice,i) => (voiceselect.options[i] = new
Option(voice.name,i)));
};

voiceselect.addEventListener("change", () => {
    speech.voice = voices[voiceselect.value];
});



document.querySelector("button").addEventListener("click",() => {
    speech.text = document.querySelector("textarea").value;
    window.speechSynthesis.speak(speech);
});
```

B. RESULT


Text To Speech Converter
Type Here....
Microsoft David - English (United States)
play


Text To Speech Converter
HELLO EVERYONE ...! THIS IS C.S.S.SASTIKRUTHIGA.,
FROM III BSC,COMPUTER SCIENCE
THIS IS A PROJECT FOR CONVERTING TEXT TO SPEECH
WHICH CONVAY THE TEXT IN FORM OF AUDIO
Microsoft David - English (United States)
play