

Text To Speech Converter

The image shows a web application interface for a Text To Speech Converter. It features a white rounded rectangle centered on a solid blue background. Inside the white rectangle, the title "Text To Speech" is displayed in a large, black, sans-serif font. Below the title, the label "Enter Text" is positioned above a large, empty text input field. Further down, the label "Select Voice" is placed above a dropdown menu. The dropdown menu is open, showing the selected option "Microsoft David - English (United States) (en-US)" with a small downward arrow to its right. At the bottom of the white rectangle is a prominent blue button with the text "Convert To Speech" in white, sans-serif font.

Project Link

<https://sandeep-balli.github.io/Text-To-Speech-Converter/>

Project Description

Text to Speech Converter is a web application that helps us to convert our text into speech in different voices depending on the system.

This project is purely based on SpeechSynthesis Web API.

If the length of the text is above 80 letters, the pause and resume functionality will be enabled.

Languages Used

HTML, CSS, JavaScript

Code Workflow

```
function voices() {
  for (let voice of synth.getVoices()) {
    let selected = voice.name === "Google US English" ? "selected" : "";
    let option = `<option value="${voice.name}" ${selected}>${voice.name} (${voice.lang})
  </option>`;
    voiceList.insertAdjacentHTML("beforeend", option); //Inserting option tag beforeend
  }
}
```

In the above code, `synth.getVoices()` returns a list of voices that are available on the current device.

We are selecting the “Google US English” as the default language.

We are creating an option tag with passing voice name and voice language.

```
function textToSpeech(text) {
  let utterance = new SpeechSynthesisUtterance(text);
  for (let voice of synth.getVoices()) {
    if (voice.name === voiceList.value) {
      utterance.voice = voice;
    }
  }
  synth.speak(utterance);
}
```

The above code does the conversion part of the project.

Here, `SpeechSynthesisUtterance()` represents a speech request. It contains the content the speech service should read and information about how to read it (language, pitch and volume).

```

speechBtn.addEventListener("click", e => {
  e.preventDefault();
  if (textArea.value !== "") {
    if (!synth.speaking) {
      textToSpeech(textArea.value);
    }
    if (textArea.value.length > 80) {
      if (isSpeaking) {
        synth.resume();
        isSpeaking = false;
        speechBtn.innerText = "Pause Speech";
      } else {
        synth.pause();
        isSpeaking = true;
        speechBtn.innerText = "Resume Speech";
      }

      setInterval(() => {
        if(!synth.speaking && !isSpeaking) {
          isSpeaking = true;
          speechBtn.innerText = "Convert To Speech";
        }
      }, 1000);
    } else {
      speechBtn.innerText = "Convert To Speech";
    }
  }
})

```

The above code does the functionality of Pause and Resume.