PROJECT STRUCTURE

TEMPLATES:-

TEXT-MP3

|  |  |
| --- | --- |
| Team ID | PNT27022TMID22447 |
| Project Name | Real Time Communication System Powered by AI For Specially Abled |

TTS.HTML:

<!DOCTYPE html>

<!-- Coding By CodingNepal - youtube.com/codingnepal -->

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Text To Speech |VICTORY TTS </title>

<link rel="stylesheet" href="xxx.css">

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

</head>

<body>

<div class="wrapper">

<header>Text To Speech</header>

<form action="#">

<div class="row">

<label>Enter Text</label>

<textarea></textarea>

</div>

<div class="row">

<label>Select Voice</label>

<div class="outer">

<select></select>

</div>

</div>

<button>Convert To Speech</button>

</form>

</div>

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

</iframe>

</body>

</html>

XXX.CSS:

\* Import Google Font - Poppins \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700& display=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

body{

display: flex;

align-items: center;

justify-content:center;

min-height: 100vh;

background: #07cff2;

}

::selection{

color: #492bf0;

background: #492bf0 ;

}

.wrapper{

width: 370px;

padding: 25px 30px;

border-radius: 7px;

background: #eef207;

box-shadow: 7px 7px 20px rgba(0,0,0,0.05);

}

.wrapper header{

font-size: 28px;

font-weight: 500;

text-align: center;

}

.wrapper form{

margin: 35px 0 20px;

}

form .row{

display: flex;

margin-bottom: 20px/;

flex-direction: column;

}

form .row label{

font-size: 18px;

margin-bottom: 5px;

}

form .row:nth-child(2) label{

font-size: 17px;

}

form :where(textarea, select, button){

outline: none;

width: 100%;

height: 100%;

border: none;

border-radius: 5px;

}

form .row textarea{

resize: none;

height: 110px;

font-size: 15px;

padding: 8px 10px;

border: 1px solid #999;

}

form .row textarea::-webkit-scrollbar{

width: 0px;

}

form .row .outer{

height: 47px;

display: flex;

padding: 0 10px;

align-items: center;

border-radius: 5px;

justify-content: center;

border: 1px solid #999;

}

form .row select{

font-size: 14px;

background: none;

}

form .row select::-webkit-scrollbar{

width: 8px;

}

form .row select::-webkit-scrollbar-track{

background: #fff;

}

form .row select::-webkit-scrollbar-thumb{

background: #888;

border-radius: 8px;

border-right: 2px solid #ffffff;

}

form button{

height: 52px;

color: #fff;

font-size: 17px;

cursor: pointer;

margin-top: 10px;

background: #6347e2;

transition: 0.3s ease;

}

form button:hover{

background: #68b2dd;

}

@media(max-width: 400px){

.wrapper{

max-width: 345px;

width: 100%;

}

}

SCRIPT.JS:

const textarea = document.querySelector("textarea"),

voiceList = document.querySelector("select"),

speechBtn = document.querySelector("button");

let synth = speechSynthesis,

isSpeaking = true;

voices();

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);

}

}

synth.addEventListener("voiceschanged", voices);

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);

}

speechBtn.addEventListener("click", e =>{

e.preventDefault();

if(textarea.value !== ""){

if(!synth.speaking){

textToSpeech(textarea.value);

}

if(textarea.value.length > 80){

setInterval(()=>{

if(!synth.speaking && !isSpeaking){

isSpeaking = true;

speechBtn.innerText = "Convert To Speech";

}else{

}

}, 500);

if(isSpeaking){

synth.resume();

isSpeaking = false;

speechBtn.innerText = "Pause Speech";

}else{

synth.pause();

isSpeaking = true;

speechBtn.innerText = "Resume Speech";

}

}else{

speechBtn.innerText = "Convert To Speech";

}

}

});

LANGUAGE.JS:

var langs =

[['Afrikaans',       ['af-ZA']],

 ['Bahasa Indonesia',['id-ID']],

 ['Bahasa Melayu',   ['ms-MY']],

 ['Català',          ['ca-ES']],

 ['Čeština',         ['cs-CZ']],

 ['Deutsch',         ['de-DE']],

 ['English',         ['en-AU', 'Australia'],

                     ['en-CA', 'Canada'],

                     ['en-IN', 'India'],

                     ['en-NZ', 'New Zealand'],

                     ['en-ZA', 'South Africa'],

                     ['en-GB', 'United Kingdom'],

                     ['en-US', 'United States']],

 ['Español',         ['es-AR', 'Argentina'],

                     ['es-BO', 'Bolivia'],

                     ['es-CL', 'Chile'],

                     ['es-CO', 'Colombia'],

                     ['es-CR', 'Costa Rica'],

                     ['es-EC', 'Ecuador'],

                     ['es-SV', 'El Salvador'],

                     ['es-ES', 'España'],

                     ['es-US', 'Estados Unidos'],

                     ['es-GT', 'Guatemala'],

                     ['es-HN', 'Honduras'],

                     ['es-MX', 'México'],

                     ['es-NI', 'Nicaragua'],

                     ['es-PA', 'Panamá'],

                     ['es-PY', 'Paraguay'],

                     ['es-PE', 'Perú'],

                     ['es-PR', 'Puerto Rico'],

                     ['es-DO', 'República Dominicana'],

                     ['es-UY', 'Uruguay'],

                     ['es-VE', 'Venezuela']],

 ['Euskara',         ['eu-ES']],

 ['Français',        ['fr-FR']],

 ['Galego',          ['gl-ES']],

 ['Hrvatski',        ['hr\_HR']],

 ['IsiZulu',         ['zu-ZA']],

 ['Íslenska',        ['is-IS']],

 ['Italiano',        ['it-IT', 'Italia'],

                     ['it-CH', 'Svizzera']],

 ['Magyar',          ['hu-HU']],

 ['Nederlands',      ['nl-NL']],

 ['Norsk bokmål',    ['nb-NO']],

 ['Polski',          ['pl-PL']],

 ['Português',       ['pt-BR', 'Brasil'],

                     ['pt-PT', 'Portugal']],

 ['Română',          ['ro-RO']],

 ['Slovenčina',      ['sk-SK']],

 ['Suomi',           ['fi-FI']],

 ['Svenska',         ['sv-SE']],

 ['Türkçe',          ['tr-TR']],

 ['български',       ['bg-BG']],

 ['Pусский',         ['ru-RU']],

 ['Српски',          ['sr-RS']],

 ['한국어',            ['ko-KR']],

 ['中文',             ['cmn-Hans-CN', '普通话 (中国大陆)'],

                     ['cmn-Hans-HK', '普通话 (香港)'],

                     ['cmn-Hant-TW', '中文 (台灣)'],

                     ['yue-Hant-HK', '粵語 (香港)']],

 ['日本語',           ['ja-JP']],

 ['Lingua latīna',   ['la']]];

let select\_language = document.querySelector('#select\_language');

let select\_dialect = document.querySelector('#select\_dialect');

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

  select\_language.options[i] = new Option(langs[i][0], i);

}

select\_language.selectedIndex = 6;

updateCountry();

select\_dialect.selectedIndex = 6;

function updateCountry() {

  for (var i = select\_dialect.options.length - 1; i >= 0; i--) {

    select\_dialect.remove(i);

  }

  var list = langs[select\_language.selectedIndex];

  for (var i = 1; i < list.length; i++) {

    select\_dialect.options.add(new Option(list[i][1], list[i][0]));

  }

  select\_dialect.style.visibility = list[1].length == 1 ? 'hidden' : 'visible';

}