

Title: Text-to-Speech Web Application

A simple web app using Web
Speech API

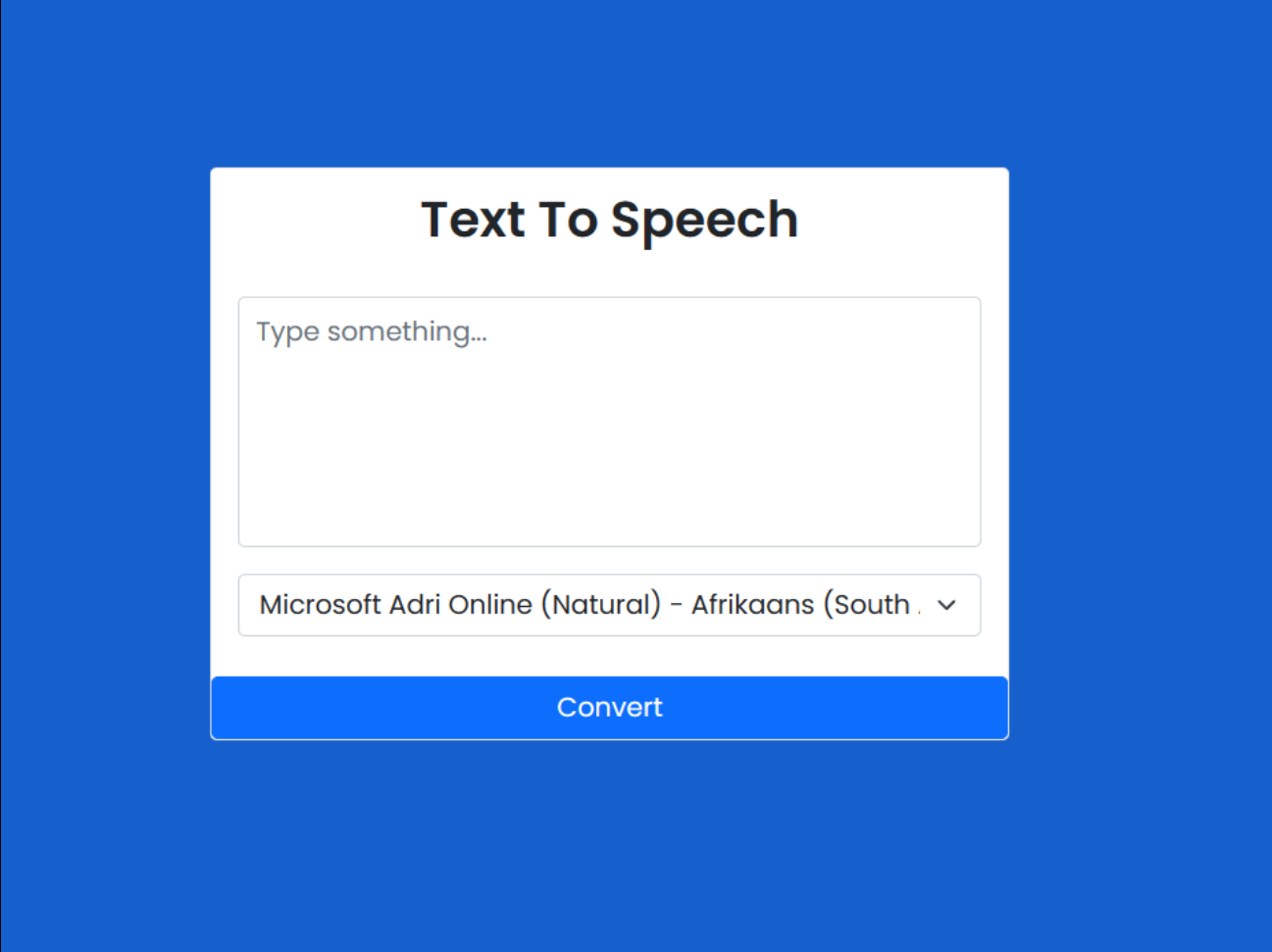
By → AMAN THAKUR

SECTION → C2

ROLL NO → 14(2218331)

Brief explanation of the purpose of the web application

The purpose of the web application is to provide a user-friendly interface for converting text into speech. Users can input text into a text area, and the application utilizes the Web Speech API to convert this text into spoken words. The Web Speech API is a browser-based API that allows developers to integrate speech recognition and synthesis capabilities into web applications. In this case, the focus is on text-to-speech conversion, enabling users to hear the content they input. This functionality can be particularly useful for accessibility purposes, language learning applications, or any scenario where spoken output enhances the user experience. The Web Speech API simplifies the process of incorporating speech synthesis into web applications, making it accessible and straightforward for developers to implement text-to-speech features.

A screenshot of a web application titled "Text To Speech" displayed on a blue background. The application interface is a white rectangular box. At the top, the title "Text To Speech" is centered in a bold, black font. Below the title is a large, empty text input area with a light gray border and the placeholder text "Type something...". Underneath the text area is a dropdown menu showing the selected voice as "Microsoft Adri Online (Natural) - Afrikaans (South)" with a downward arrow icon. At the bottom of the interface is a solid blue button with the word "Convert" written in white text.

Text To Speech

Type something...

Microsoft Adri Online (Natural) - Afrikaans (South) ▾

Convert

HTML Structure Overview :



Introduction:

Brief explanation of the HTML structure.

Highlighting the importance of a well-organized structure for web applications.



Form Element:

Description of the **<form>** element.

Mention of its role in encapsulating the input elements and handling user interactions.



Card Element:

Explanation of the **<div class="card">** element.

Emphasize its usage for grouping related content and providing a structured layout.



Text area Element:

Details about the **<textarea>** element.

Emphasis on its role as the user input field for typing text.

- The HTML structure is designed to create a visually appealing and user-friendly interface.
- The **<form>** element wraps the content, providing a logical grouping and facilitating user interactions.
- The use of the **<div class="card">** element enhances the layout and organization of the page, making it more structured and visually appealing.
- The **<textarea>** element is the primary input field where users can type the text they want to convert to speech.

CSS Structure Overview :



- **Introduction:**

- Brief explanation of the role of CSS in styling web applications.
- Highlighting the importance of visual appeal and user experience.

- **Poppins Font:**

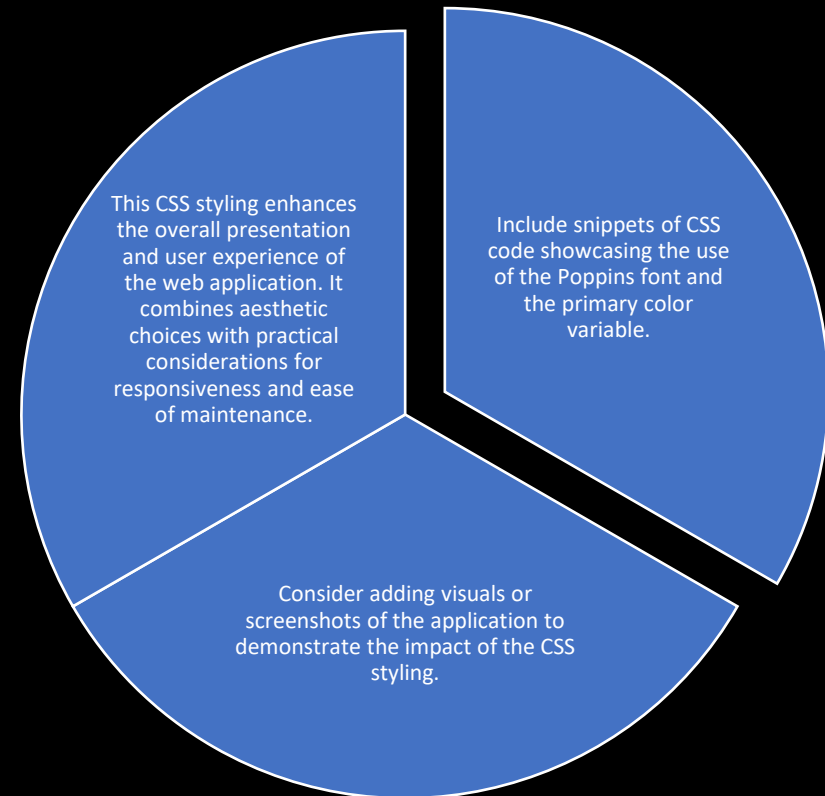
- Introduction to the use of the Poppins font.
- Mention of its modern and clean appearance, enhancing readability.

- **Primary Color Variable:**

- Explanation of the `--primary` color variable.
- Emphasis on its role in maintaining consistency and ease of color management.

- **Layout and Bootstrap Classes:**

- Overview of the layout design.
- Explanation of the use of Bootstrap classes for styling elements.



JavaScript Overview :

1.Introduction:

- Brief overview of the role of JavaScript in web development.
- Emphasis on its capability to add interactivity and dynamic behavior to web pages.

2.script.js File:

- Explanation of the **script.js** file as the JavaScript file for the web application.
- Mention of its crucial role in handling user interactions and controlling the text-to-speech functionality.

3.Variable Declarations:

- Declaration of key variables:
 - textarea**: Represents the textarea element for user input.
 - voicearea**: Represents the dropdown for selecting voices.
 - speakbutton**: Represents the button triggering text-to-speech.
 - synth**: Represents the SpeechSynthesis interface.
 - isspeaking**: Represents the status of speech synthesis.

4.Initialization of Web Speech API:

- Brief explanation of the **synth** variable as an instance of the Speech Synthesis interface.
- Mention of the importance of the Web Speech API for text-to-speech conversion.

• Additional Information:

- Consider including a brief code snippet showcasing the variable declarations and the initialization of the Web Speech API.
- Visual aids, such as a simplified flowchart or diagrams, can help illustrate the relationships between variables and the Web Speech API.
- This introduction sets the stage for understanding how JavaScript drives the interactivity and functionality of the text-to-speech web application. It establishes the significance of variables and the Web Speech API in achieving the desired user experience.

Voice Function Explanation :

1.Voices Function:

- Introduction to the **voices** function in script.js.
- Explanation of its role in populating the voice selection dropdown.

2.Triggering Event:

- Mention of the "voiceschanged" event as the trigger for the **voices** function.
- Explanation that this event is fired when the list of available voices changes.

3.Voice Population:

- Overview of the loop within the **voices** function.
- For each available voice, an option is created for the voice selection dropdown.
- Information displayed includes the voice name and language.

4.Default Selection:

- Mention of the conditional check for the "Google US English" voice.
- Explanation that this voice is set as the default selected option.



textToSpeak Function Explanation :

1.textToSpeak Function:

- Introduction to the **textToSpeak** function in script.js.
- Explanation of its role in initiating text-to-speech conversion.

2.SpeechSynthesisUtterance Object:

- Overview of the **SpeechSynthesisUtterance** object.
- Mention that it represents the text to be spoken and includes properties like pitch and rate.

3.Voice Selection:

- Description of the loop within the **textToSpeak** function.
- For each available voice, a check is performed to match the selected voice from the voice selection dropdown.
- The chosen voice is then assigned to the **SpeechSynthesisUtterance** object.

4.Synthesis Invocation:

- Explanation that the **synth.speak(utter)** command initiates the text-to-speech synthesis.





Button Clicked Overview :

- **Introduction:**
 - The button click event listener manages the initiation and control of text-to-speech conversion in the application.
- **Text-to-Speech Conversion:**
 - Clarify that the text-to-speech conversion is triggered only if the textarea has content.
- **Input Length Check:**
 - Explain that the behavior of the button changes based on the length of the entered text.
 - Different actions are taken for short and long text inputs.
- **Pause and Resume Functionality:**
 - Describe the logic for toggling between pause and resume based on the current state.
 - Users can pause, resume, or convert text to speech.
- **Convert Functionality:**
 - Highlight the dynamic nature of the button label, changing between "Pause," "Resume," and "Convert" based on the text length and synthesis state.

Advantages :

User-Friendly Interface:

- Intuitive and easy-to-navigate design.
- Clean layout with a card element and structured form.
- Logical flow of information for a seamless user experience.
- Integration of Bootstrap for consistent and polished styling.

Voice Selection Options:

- Voice selection dropdown for personalized text-to-speech experience.
- Dynamic population of the dropdown based on the "voiceschanged" event.
- Default voice ("Google US English") for a sensible starting point.

Pause and Resume Functionality:

- Interactive controls for pausing and resuming text-to-speech synthesis.
- Adaptive button behavior based on the current state.
- Users can easily toggle between pause and resume.

Responsive Design:

- Layout adaptability for different screen sizes.
- Utilization of Bootstrap framework for responsive styling.
- Consistent user experience across various devices.



Conclusion-

- The text-to-speech web application provides a seamless and user-friendly experience with its clean layout, logical flow, and Bootstrap integration.
- Users have the flexibility to personalize their experience with voice selection options, benefiting from the dynamic population of the dropdown.
- Interactive controls for pause and resume functionality enhance user control, adapting to individual preferences during text-to-speech synthesis.
- The application's responsive design ensures a consistent and positive user experience across a variety of devices, making it accessible and user-friendly.
- In summary, the text-to-speech web application successfully combines a user-centric interface, voice customization options, interactive functionality, and responsive design, delivering a versatile and enjoyable experience for users.
-

