PROJECT REPORT

on

Text-To-Voice Convertor using HTML,CSS and JavaScript

(CSE III Semester Mini project)

2023-2024

Submitted to:

Mr Ajay Shukla

(CC-CSE-C2-III-Sem)

Submitted by:

Mr Aman Thakur

UNIVERSITY.NO

→2218331(14)

CSE-C2-III-Sem

Session: 2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

Certified that Mr. Aman Thakur (Roll No.- 2218331) has developed mini project on “Text to voice convertor” for the CS III Semester Mini Project Lab in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date: 20/01/2024

Mr Ajay Shukla

Class Co-ordinator

CSE-C2-III-Sem

(CSE Department)

GEHU Dehradun

ACKNOWLEDGMENT

We would like to express our gratitude to GOD, the most Beneficent and the most Merciful, for completion of project.

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-ordinator Mr Ajay Shukla for his patience, support and encouragement throughout the completion of this project and having faith in us.

At last but not the least We greatly indebted to all other persons who directly or indirectly helped us during this work.

Mr. Aman Thakur

Roll No.- 2218331

Class Roll No - 14

CSE-C2-III-Sem

Session: 2023-2024

GEHU, Dehradun

TABLE OF CONTENTS

1. INTRODUCTION

Web development is a dynamic and rapidly evolving field that encompasses the creation and maintenance of websites and web applications. In today's digital age, a strong online presence is crucial for businesses, organizations, and individuals. Web development plays a pivotal role in shaping the digital landscape, providing interactive and user-friendly experiences for a diverse audience.

Key Components of Web Development:

Front-End Development: Front-end development involves designing and implementing the visual aspects of a website or web application that users interact with directly. It encompasses HTML, CSS, and JavaScript to create responsive and engaging user interfaces. Modern front-end development often involves the use of frameworks and libraries like React, Angular, or Vue.js to streamline the development process and enhance user experiences.

Back-End Development: Back-end development focuses on server-side logic, databases, and application functionality that users don't see. Common back-end programming languages include Python, Ruby, PHP, and Node.js. Developers use frameworks such as Django, Ruby on Rails, Laravel, or Express.js to efficiently build robust and scalable server-side applications.

Full-Stack Development: Full-stack developers have expertise in both front-end and back-end development, allowing them to work on all aspects of a web application. This comprehensive skill set enables them to understand the entire development process, from user interface design to server-side implementation and database management.

Web Development Tools: Web developers use a variety of tools and technologies to streamline their workflow. Integrated Development Environments (IDEs) like Visual Studio Code, Sublime Text, and Atom provide a conducive environment for coding. Version control systems like Git help manage and track

changes collaboratively, ensuring a seamless development process.

Responsive Design: With the increasing use of various devices such as smartphones, tablets, and desktops, responsive web design has become crucial. Developers employ techniques like media queries and flexible grids to ensure that websites and applications adapt seamlessly to different screen sizes, providing a consistent user experience across devices.

Web Security: As cyber threats continue to evolve, web security has become a paramount concern. Developers need to implement secure coding practices, use encryption protocols (HTTPS), and regularly update dependencies to protect websites and applications from potential vulnerabilities.

Conclusion: Web development is a multifaceted discipline that continues to evolve alongside technological advancements. As user expectations grow, web developers play a crucial role in creating innovative, secure, and user-friendly online experiences. Staying abreast of emerging technologies and best

practices is essential for professionals in this field to contribute effectively to the ever-changing landscape of the internet.

Title: An Overview of HTML and Its Important Tags

Introduction: HTML (Hypertext Markup Language) is the standard markup language used to create the structure of web pages. It provides a framework for organizing content and enables the creation of documents that can be displayed in web browsers. HTML uses a system of tags to define and categorize different elements on a webpage.

Important HTML Tags:

1. **<!DOCTYPE html>:** This declaration at the beginning of an HTML document defines the document type and version. It ensures that the browser renders the page correctly.
2. **<html>:** The root element of an HTML document, encapsulating the entire content.
3. **<head>:** This section contains meta-information about the document, such as the title, character set, and linked stylesheets or scripts.

4. **<title>:** Specifies the title of the webpage, which appears in the browser's title bar or tab.
5. **<body>:** Encloses the content of the webpage, including text, images, links, and other elements visible to users.
6. **<h1> to <h6>:** Heading tags define headings of varying levels. **<h1>** is the highest level (main heading), and **<h6>** is the lowest level.
7. **<p>:** Defines paragraphs of text, creating breaks before and after the content.
8. **<a>:** Creates hyperlinks, allowing users to navigate to other pages or resources.
9. **<img>:** Embeds images into the webpage, enhancing visual content.
10. **<ul>, <ol>, <li>:** Unordered lists (**<ul>**), ordered lists (**<ol>**), and list items (**<li>**) structure and organize content in bullet points or numbered lists.

11. ** and :** These container tags are used to group and style elements. ** ** is a block-level container, and **** is an inline container.

12. **<table>, <tr>, <td>:** Table tags are used to create tabular data. **<tr>** represents a table row, and **<td>** represents a table cell.

13. **<form>, <input>, <button>:** Form tags allow the creation of interactive elements such as input fields and buttons for user input.

14. **<script>:** Embeds JavaScript code within the HTML document, enabling client-side scripting and interactivity.

15. **<meta>:** Defines metadata, including character set, viewport settings, and other information for browsers and search engines.

```
<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Text To Speech</title>
    <!-- Bootstrap CSS -->
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" >
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div class="wrapper">
        <form action="">
            <div class="card">
                <div class="card-header">
                    <p>Text To Speech</p>
                </div>
                <div class="card-body mb-2">
                    <div class="user-text mb-3">
                        <textarea class="form-control" name="" id="" placeholder="Type something..."></textarea>
                    </div>
                    <div class="select-language">
                        <select class="form-select" name="" id="">
                        </select>
                    </div>
                </div>
                <button class="btn btn-primary" id="button">Convert</button>
            </div>
        </form>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

Title: Essential Overview of CSS and Key Styling Properties

Introduction: CSS (Cascading Style Sheets) is a styling language used in web development to control the presentation and layout of HTML documents. It enables developers to enhance the visual appeal of web pages by defining styles such as colors, fonts, spacing, and positioning.

Key CSS Concepts:

1. **Selectors:** CSS selectors are patterns that define which HTML elements the styles will be applied to. They can target specific elements, classes, IDs, or even attributes.
2. **Properties:** CSS properties are the styling attributes applied to selected elements. Each property has a specific value that defines how the styling feature should be presented.
3. **Color:** The **color** property sets the text color, and the **background-color** property determines the background color of an element. Colors can be specified using names, hex codes, RGB, or HSL values.

4. **Typography:** CSS controls text styling through properties such as **font-family** (defining the font), **font-size** (setting the text size), and **font-weight** (adjusting the thickness of characters).

5. **Layout:** Properties like **margin**, **padding**, **display**, and **position** are used to control the spacing, layout, and positioning of elements on the page.

6. **Box Model:** The box model comprises properties like **width**, **height**, **margin**, **padding**, and **border**, defining the dimensions and spacing around an element.

7. **Flexbox and Grid:** CSS provides layout models like Flexbox and Grid, offering efficient ways to create responsive and dynamic layouts for complex web designs.

8. **Transitions and Animations:** CSS enables smooth transitions between styles using the **transition** property and allows the creation of animations through the **@keyframes** rule.

9. **Media Queries:** Media queries allow developers to apply different styles based on the characteristics of the device, such

as screen width, height, or orientation, facilitating responsive design.

10. **Selectors Combinators:** CSS selectors can be combined to target specific elements more precisely. Combinators include the descendant selector (**space**), child selector (>), and sibling selectors (+ and ~).

PROJECT CODE -

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500
;600;700;800;900&display=swap');
*{
    font-family: 'Poppins', sans-serif;
}
:root{
    --primary: #1660ce;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: var(--primary);
}

.wrapper{
    width: 30rem;
}

.card-header {
    background-color: #fff;
    border-bottom: none;
    text-align: center;
}
.card-header p{
```

```
    font-size: 30px;
    font-weight: 600;
    margin-bottom: 0;
}

.card .card-body .user-text textarea{
    resize: unset;
    height: 150px;
    padding: 8px 10px;
    font-size: 16px;
}

.card button {
    display: flex;
    align-items: center;
    justify-content: center;
}
```

Title: JavaScript Essentials: A Brief Overview of Core Concepts and Functionality

Introduction: JavaScript is a versatile scripting language primarily used to enhance the interactivity and dynamic behavior of web pages. As a client-side programming language, JavaScript enables developers to create responsive and engaging user interfaces.

Key JavaScript Concepts:

1. **Variables:** Variables are used to store and manage data. They are declared using the **var**, **let**, or **const** keyword and can hold various types of values, including numbers, strings, and objects.
2. **Data Types:** JavaScript supports various data types, such as numbers, strings, booleans, arrays, and objects. Understanding data types is crucial for effective data manipulation.
3. **Functions:** Functions are blocks of reusable code that perform specific tasks. They are defined using the **function** keyword and can be called to execute their code.

4. **Conditional Statements:** JavaScript includes **if**, **else if**, and **else** statements for decision-making. These control the flow of the program based on specified conditions.

5. **Loops:** Loops, like **for** and **while**, allow developers to execute a block of code repeatedly. They are useful for iterating through arrays, objects, or performing operations a certain number of times.

6. **Arrays:** Arrays are ordered lists that can hold multiple values. JavaScript provides methods for manipulating arrays, such as **push**, **pop**, **shift**, and **unshift**.

7. **Objects:** Objects group related data and functions. They consist of key-value pairs and are fundamental to JavaScript's object-oriented programming paradigm.

8. **DOM Manipulation:** The Document Object Model (DOM) represents the structure of an HTML document. JavaScript can manipulate the DOM to dynamically update and modify the content and style of web pages.

9. **Event Handling:** JavaScript enables the creation of interactive web pages by responding to user actions, such as clicks, keypresses, or mouse movements, through event handling.

10. **AJAX (Asynchronous JavaScript and XML):** AJAX allows data to be retrieved from a server asynchronously, without reloading the entire page. It is commonly used in modern web applications to enhance user experience.

11. **ES6 Features:** Modern JavaScript, known as ECMAScript 6 (ES6), introduced features like arrow functions, template literals, destructuring, and the **let** and **const** keywords for more efficient and concise coding.

12. **Promises and Async/Await:** JavaScript uses promises to handle asynchronous operations, and ES6 introduced the **async** and **await** keywords, simplifying asynchronous code and making it more readable.

PROJECT CODE –

```
const textarea = document.querySelector("textarea"),
voiceList = document.querySelector("select"),
speakButton = document.querySelector("button");
```

```
let synth = speechSynthesis;
let isSpeaking = true;

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 textToSpeak(text) {
    let utter = new SpeechSynthesisUtterance(text);

    for(let voice of synth.getVoices()){
        if(voice.name === voiceList.value){
            utter.voice = voice;
        }
    }

    synth.speak(utter);
}

speakButton.addEventListener("click", e => {
    e.preventDefault();
    if(textarea.value !== ""){
        if(!synth.speaking){
            textToSpeak(textarea.value);
        }

        if(textarea.value.length > 80) {
            if(isSpeaking){
                synth.resume();
                isSpeaking = false;
                speakButton.innerText = "Pause";
            }
            else{
                synth.pause();
                isSpeaking = true;
                speakButton.innerText = "Resume";
            }
            setInterval( () =>{
                if(!synth.speaking && !isSpeaking){
                    isSpeaking = true;
                    speakButton.innerText = "Convert";
```

```
                }
            });
        } else {
            speakButton.innerText = "Convert";
        }
    }
});
```

ABOUT THE PROJECT →

1. **User-Friendly Interface:**
    - **The design is simple and user-friendly, with a clean layout that includes a card element and a form for input.**
2. **Voice Selection:**
    - **Users can select their preferred voice from the dropdown, allowing for customization of the speech output.**
3. **Pause and Resume Functionality:**

- **The application allows users to pause and resume the speech synthesis, providing more control over the listening experience.**

4. **Responsive Design:**

    - **The application is designed to be responsive, making it suitable for various screen sizes.**

5. **Integration of Bootstrap:**

    - **The use of Bootstrap classes enhances the visual appeal and responsiveness of the application without the need for extensive custom styling.**

6. **Voice Options:**

    - **The code dynamically populates the voice selection dropdown with available voices, allowing users to choose from a list.**

7. **Sensible Default:**

- **The application defaults to the "Google US English" voice, providing a sensible choice for users who may not want to explore voice options.**

8. **Dynamic Voice Population:**

    - **The code dynamically updates the voice selection dropdown when the "voiceschanged" event is triggered, ensuring that users have access to the latest available voices.**

REFERENCE –

YouTube and google