HTML CODE:

<

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Your Website Title</title>

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

</head>

<body>

       <video autoplay muted loop id="bg-video">

        <source src="C:\Users\91972\OneDrive\Desktop\New folder\Practice\47717-451812833\_medium.mp4" type="video/mp4">

        Your browser does not support HTML5 video.

      </video>

    <header>

        <h1>Resume Verifier</h1>

    </header>

    <nav>

        <div>

            <a href="#" data-title="Home">Home</a>

            <a href="#" data-title="About">About</a>

            <a href="#" data-title="Service">Service</a>

            <a href="#" data-title="Contact">Contact</a>

        </div>

    </nav>

    <br/>

    <br/>

    <section>

        <form>

            <div class="job-description">

                <label for="jobDescription">Job description:</label>

                <textarea id="jobDescription" name="jobDescription" rows="4" cols="50"></textarea>

            </div>

            <div class="file-upload">

                <label for="pdfFile">Upload PDF:</label>

                <input type="file" id="pdfFile" name="pdfFile">

            </div>

            <br/>

            <br/>

            <div class="button-container">

                <input type="button" value="Generate Score" onclick="generateScore()">

                <input type="reset" value="Reset">

            </div>

        </form>

    </section>

    <footer>

        <p id="scoreResult"></p>

    </footer>

    <!-- Include PDF.js library -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.9.359/pdf.min.js"></script>

    <!-- JavaScript code -->

    <script>

        function extractTextFromPDF(pdfData) {

            return new Promise((resolve, reject) => {

                const typedArray = new Uint8Array(pdfData);

                pdfjsLib.getDocument({ data: typedArray }).promise.then(pdf => {

                    let fullText = '';

                    const numPages = pdf.numPages;

                    const getPageText = async (pageNumber) => {

                        const page = await pdf.getPage(pageNumber);

                        const textContent = await page.getTextContent();

                        const pageText = textContent.items.map(item => item.str).join(' ');

                        return pageText;

                    };

                    const getTextFromAllPages = async () => {

                        for (let i = 1; i <= numPages; i++) {

                            const pageText = await getPageText(i);

                            fullText += pageText + ' ';

                        }

                        resolve(fullText);

                    };

                    getTextFromAllPages();

                }).catch(error => {

                    reject(error);

                });

            });

        }

        function extractKeywords(text) {

            const stopwords = [

                'the', 'and', 'is', 'in', 'to', 'of', 'for', 'with', 'a', 'that', 'it', 'on', 'as',

                'an', 'at', 'this', 'be', 'are', 'was', 'or', 'by', 'from', 'but', 'not', 'have',

                'has', 'been', 'such', 'which', 'also', 'can', 'may', 'will', 'must', 'should', 'could',

                'would', 'might', 'their', 'there', 'then', 'than',

                'do', 'does', 'doing', 'done', 'get', 'getting', 'got', 'been', 'being', 'make', 'makes',

                'making', 'made', 'see', 'seen', 'saw', 'look', 'looks', 'looking', 'take', 'takes', 'taking',

                'took', 'think', 'thinks', 'thinking', 'thought', 'use', 'uses', 'using', 'used', 'want',

                'wants', 'wanted', 'try', 'tries', 'trying', 'tried', 'come', 'comes', 'coming', 'came',

                'good', 'better', 'best', 'bad', 'worse', 'worst', 'great', 'greater', 'greatest', 'high',

                'higher', 'highest', 'low', 'lower', 'lowest', 'big', 'bigger', 'biggest', 'small', 'smaller',

                'smallest', 'new', 'old', 'young', 'happy', 'happier', 'happiest', 'sad', 'sadder', 'saddest'

            ];

            const words = text.toLowerCase().match(/\b\w+\b/g).filter(word => !stopwords.includes(word));

            const wordFrequency = {};

            words.forEach(word => {

                if (wordFrequency[word]) {

                    wordFrequency[word]++;

                } else {

                    wordFrequency[word] = 1;

                }

            });

            const sortedKeywords = Object.keys(wordFrequency).sort((a, b) => wordFrequency[b] - wordFrequency[a]);

            return sortedKeywords.slice(0, 10);

        }

        function compareKeywords(jobKeywords, resumeKeywords) {

            const uniqueKeywords = new Set([...jobKeywords, ...resumeKeywords]);

            const commonKeywords = jobKeywords.filter(word => resumeKeywords.includes(word));

            const score = (commonKeywords.length / uniqueKeywords.size) \* 100;

            return score.toFixed(2);

        }

        function generateScore() {

            var jobDescription = document.getElementById("jobDescription").value.toLowerCase();

            var fileInput = document.getElementById('pdfFile');

            var file = fileInput.files[0];

            var reader = new FileReader();

            reader.onload = function(e) {

                extractTextFromPDF(e.target.result)

                    .then(pdfText => {

                        var resumeKeywords = extractKeywords(pdfText);

                        var jobKeywords = extractKeywords(jobDescription);

                        var comparisonScore = compareKeywords(jobKeywords, resumeKeywords);

                        document.getElementById("scoreResult").innerHTML = "Matching Score: " + comparisonScore + "%";

                    })

                    .catch(error => {

                        console.error('Error extracting text from PDF:', error);

                    });

            };

            reader.readAsArrayBuffer(file);

        }

    </script>

</body>

</html>

CSS code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Your Website Title</title>

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

</head>

<body>

       <video autoplay muted loop id="bg-video">

        <source src="C:\Users\91972\OneDrive\Desktop\New folder\Practice\47717-451812833\_medium.mp4" type="video/mp4">

        Your browser does not support HTML5 video.

      </video>

    <header>

        <h1>Resume Verifier</h1>

    </header>

    <nav>

        <div>

            <a href="#" data-title="Home">Home</a>

            <a href="#" data-title="About">About</a>

            <a href="#" data-title="Service">Service</a>

            <a href="#" data-title="Contact">Contact</a>

        </div>

    </nav>

    <br/>

    <br/>

    <section>

        <form>

            <div class="job-description">

                <label for="jobDescription">Job description:</label>

                <textarea id="jobDescription" name="jobDescription" rows="4" cols="50"></textarea>

            </div>

            <div class="file-upload">

                <label for="pdfFile">Upload PDF:</label>

                <input type="file" id="pdfFile" name="pdfFile">

            </div>

            <br/>

            <br/>

            <div class="button-container">

                <input type="button" value="Generate Score" onclick="generateScore()">

                <input type="reset" value="Reset">

            </div>

        </form>

    </section>

    <footer>

        <p id="scoreResult"></p>

    </footer>

    <!-- Include PDF.js library -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.9.359/pdf.min.js"></script>

    <!-- JavaScript code -->

    <script>

        function extractTextFromPDF(pdfData) {

            return new Promise((resolve, reject) => {

                const typedArray = new Uint8Array(pdfData);

                pdfjsLib.getDocument({ data: typedArray }).promise.then(pdf => {

                    let fullText = '';

                    const numPages = pdf.numPages;

                    const getPageText = async (pageNumber) => {

                        const page = await pdf.getPage(pageNumber);

                        const textContent = await page.getTextContent();

                        const pageText = textContent.items.map(item => item.str).join(' ');

                        return pageText;

                    };

                    const getTextFromAllPages = async () => {

                        for (let i = 1; i <= numPages; i++) {

                            const pageText = await getPageText(i);

                            fullText += pageText + ' ';

                        }

                        resolve(fullText);

                    };

                    getTextFromAllPages();

                }).catch(error => {

                    reject(error);

                });

            });

        }

        function extractKeywords(text) {

            const stopwords = [

                'the', 'and', 'is', 'in', 'to', 'of', 'for', 'with', 'a', 'that', 'it', 'on', 'as',

                'an', 'at', 'this', 'be', 'are', 'was', 'or', 'by', 'from', 'but', 'not', 'have',

                'has', 'been', 'such', 'which', 'also', 'can', 'may', 'will', 'must', 'should', 'could',

                'would', 'might', 'their', 'there', 'then', 'than',

                'do', 'does', 'doing', 'done', 'get', 'getting', 'got', 'been', 'being', 'make', 'makes',

                'making', 'made', 'see', 'seen', 'saw', 'look', 'looks', 'looking', 'take', 'takes', 'taking',

                'took', 'think', 'thinks', 'thinking', 'thought', 'use', 'uses', 'using', 'used', 'want',

                'wants', 'wanted', 'try', 'tries', 'trying', 'tried', 'come', 'comes', 'coming', 'came',

                'good', 'better', 'best', 'bad', 'worse', 'worst', 'great', 'greater', 'greatest', 'high',

                'higher', 'highest', 'low', 'lower', 'lowest', 'big', 'bigger', 'biggest', 'small', 'smaller',

                'smallest', 'new', 'old', 'young', 'happy', 'happier', 'happiest', 'sad', 'sadder', 'saddest'

            ];

            const words = text.toLowerCase().match(/\b\w+\b/g).filter(word => !stopwords.includes(word));

            const wordFrequency = {};

            words.forEach(word => {

                if (wordFrequency[word]) {

                    wordFrequency[word]++;

                } else {

                    wordFrequency[word] = 1;

                }

            });

            const sortedKeywords = Object.keys(wordFrequency).sort((a, b) => wordFrequency[b] - wordFrequency[a]);

            return sortedKeywords.slice(0, 10);

        }

        function compareKeywords(jobKeywords, resumeKeywords) {

            const uniqueKeywords = new Set([...jobKeywords, ...resumeKeywords]);

            const commonKeywords = jobKeywords.filter(word => resumeKeywords.includes(word));

            const score = (commonKeywords.length / uniqueKeywords.size) \* 100;

            return score.toFixed(2);

        }

        function generateScore() {

            var jobDescription = document.getElementById("jobDescription").value.toLowerCase();

            var fileInput = document.getElementById('pdfFile');

            var file = fileInput.files[0];

            var reader = new FileReader();

            reader.onload = function(e) {

                extractTextFromPDF(e.target.result)

                    .then(pdfText => {

                        var resumeKeywords = extractKeywords(pdfText);

                        var jobKeywords = extractKeywords(jobDescription);

                        var comparisonScore = compareKeywords(jobKeywords, resumeKeywords);

                        document.getElementById("scoreResult").innerHTML = "Matching Score: " + comparisonScore + "%";

                    })

                    .catch(error => {

                        console.error('Error extracting text from PDF:', error);

                    });

            };

            reader.readAsArrayBuffer(file);

        }

    </script>

</body>

</html>