<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Video Transcription Tool</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

min-height: 100vh;

display: flex;

align-items: center;

justify-content: center;

padding: 20px;

}

.container {

background: white;

border-radius: 20px;

box-shadow: 0 20px 40px rgba(0,0,0,0.1);

padding: 40px;

width: 100%;

max-width: 800px;

animation: slideUp 0.6s ease-out;

}

@keyframes slideUp {

from {

opacity: 0;

transform: translateY(30px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

.header {

text-align: center;

margin-bottom: 40px;

}

.header h1 {

color: #333;

font-size: 2.5rem;

margin-bottom: 10px;

background: linear-gradient(135deg, #667eea, #764ba2);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

background-clip: text;

}

.header p {

color: #666;

font-size: 1.1rem;

}

.form-group {

margin-bottom: 25px;

}

label {

display: block;

margin-bottom: 8px;

color: #333;

font-weight: 600;

}

input, select, textarea {

width: 100%;

padding: 15px;

border: 2px solid #e1e5e9;

border-radius: 12px;

font-size: 1rem;

transition: all 0.3s ease;

outline: none;

}

input:focus, select:focus, textarea:focus {

border-color: #667eea;

box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);

}

.btn {

background: linear-gradient(135deg, #667eea, #764ba2);

color: white;

border: none;

padding: 15px 30px;

border-radius: 12px;

font-size: 1.1rem;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

width: 100%;

position: relative;

overflow: hidden;

}

.btn:hover {

transform: translateY(-2px);

box-shadow: 0 10px 25px rgba(102, 126, 234, 0.3);

}

.btn:active {

transform: translateY(0);

}

.btn:disabled {

opacity: 0.6;

cursor: not-allowed;

transform: none;

box-shadow: none;

}

.loading {

display: none;

text-align: center;

margin: 30px 0;

}

.spinner {

border: 4px solid #f3f3f3;

border-top: 4px solid #667eea;

border-radius: 50%;

width: 40px;

height: 40px;

animation: spin 1s linear infinite;

margin: 0 auto 15px;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

.result {

margin-top: 30px;

padding: 25px;

border-radius: 12px;

display: none;

}

.result.success {

background: #f0f9f4;

border: 2px solid #10b981;

}

.result.error {

background: #fef2f2;

border: 2px solid #ef4444;

}

.result h3 {

margin-bottom: 15px;

color: #333;

}

.transcript {

background: white;

padding: 20px;

border-radius: 8px;

line-height: 1.6;

max-height: 300px;

overflow-y: auto;

border: 1px solid #e5e7eb;

}

.actions {

margin-top: 15px;

display: flex;

gap: 10px;

flex-wrap: wrap;

}

.btn-secondary {

background: #6b7280;

padding: 10px 20px;

font-size: 0.9rem;

flex: 1;

min-width: 120px;

}

.file-upload {

position: relative;

display: inline-block;

width: 100%;

}

.file-input {

position: absolute;

opacity: 0;

width: 100%;

height: 100%;

cursor: pointer;

}

.file-label {

display: block;

padding: 15px;

border: 2px dashed #e1e5e9;

border-radius: 12px;

text-align: center;

transition: all 0.3s ease;

cursor: pointer;

}

.file-label:hover {

border-color: #667eea;

background: #f8faff;

}

.progress-bar {

width: 100%;

height: 6px;

background: #e5e7eb;

border-radius: 3px;

overflow: hidden;

margin: 15px 0;

display: none;

}

.progress-fill {

height: 100%;

background: linear-gradient(90deg, #667eea, #764ba2);

transition: width 0.3s ease;

width: 0%;

}

.features {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 20px;

margin: 30px 0;

}

.feature {

text-align: center;

padding: 20px;

background: #f8faff;

border-radius: 12px;

}

.feature-icon {

font-size: 2rem;

margin-bottom: 10px;

}

@media (max-width: 768px) {

.container {

padding: 20px;

margin: 10px;

}

.header h1 {

font-size: 2rem;

}

.actions {

flex-direction: column;

}

.btn-secondary {

flex: none;

}

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<h1>🎵 Video Transcription</h1>

<p>Convert video speech to text using AI-powered transcription</p>

</div>

<div class="features">

<div class="feature">

<div class="feature-icon">🎬</div>

<h4>Multiple Platforms</h4>

<p>YouTube, TikTok, Instagram, and more</p>

</div>

<div class="feature">

<div class="feature-icon">🌍</div>

<h4>Multi-Language</h4>

<p>Support for 50+ languages</p>

</div>

<div class="feature">

<div class="feature-icon">⚡</div>

<h4>Fast Processing</h4>

<p>Quick and accurate results</p>

</div>

</div>

<form id="transcriptionForm">

<div class="form-group">

<label for="videoUrl">Video URL</label>

<input

type="url"

id="videoUrl"

placeholder="https://www.youtube.com/watch?v=..."

required

>

</div>

<div class="form-group">

<label for="language">Language</label>

<select id="language">

<option value="en-US">English (US)</option>

<option value="en-GB">English (UK)</option>

<option value="es-ES">Spanish</option>

<option value="fr-FR">French</option>

<option value="de-DE">German</option>

<option value="it-IT">Italian</option>

<option value="pt-BR">Portuguese (Brazil)</option>

<option value="ru-RU">Russian</option>

<option value="ja-JP">Japanese</option>

<option value="ko-KR">Korean</option>

<option value="zh-CN">Chinese (Simplified)</option>

<option value="ar-SA">Arabic</option>

<option value="hi-IN">Hindi</option>

</select>

</div>

<div class="form-group">

<div class="file-upload">

<input type="file" id="audioFile" class="file-input" accept="audio/\*,video/\*">

<label for="audioFile" class="file-label">

📁 Or upload audio/video file (optional)

</label>

</div>

</div>

<button type="submit" class="btn" id="submitBtn">

Start Transcription

</button>

</form>

<div class="progress-bar" id="progressBar">

<div class="progress-fill" id="progressFill"></div>

</div>

<div class="loading" id="loading">

<div class="spinner"></div>

<p>Processing your video... This may take a few minutes.</p>

</div>

<div class="result" id="result">

<h3 id="resultTitle"></h3>

<div class="transcript" id="transcript"></div>

<div class="actions">

<button class="btn btn-secondary" onclick="copyTranscript()">

📋 Copy Text

</button>

<button class="btn btn-secondary" onclick="downloadTranscript()">

💾 Download

</button>

<button class="btn btn-secondary" onclick="shareTranscript()">

🔗 Share

</button>

</div>

</div>

</div>

<script>

class WebTranscriptionService {

constructor() {

this.apiEndpoint = '/api/transcribe'; // Backend API endpoint

this.currentTranscript = '';

this.currentVideoUrl = '';

}

async transcribeVideo(videoUrl, language = 'en-US', file = null) {

const formData = new FormData();

if (file) {

formData.append('file', file);

} else {

formData.append('url', videoUrl);

}

formData.append('language', language);

try {

const response = await fetch(this.apiEndpoint, {

method: 'POST',

body: formData,

headers: {

'X-Requested-With': 'XMLHttpRequest'

}

});

if (!response.ok) {

throw new Error(`Server error: ${response.status}`);

}

const result = await response.json();

return result;

} catch (error) {

console.error('Transcription error:', error);

throw error;

}

}

// Simulated transcription for demo purposes

async simulateTranscription(videoUrl, language, file) {

return new Promise((resolve) => {

setTimeout(() => {

const sampleTranscripts = [

"Welcome to this amazing video! Today we're going to explore the fascinating world of artificial intelligence and machine learning. These technologies are revolutionizing how we interact with computers and process information.",

"In this tutorial, we'll learn how to build a modern web application using the latest technologies. We'll cover everything from setup to deployment, ensuring you have all the tools you need to succeed.",

"Music has the power to move us, inspire us, and bring us together. In this performance, we explore the emotional depths of classical composition and its relevance in today's world."

];

const randomTranscript = sampleTranscripts[Math.floor(Math.random() \* sampleTranscripts.length)];

resolve({

success: true,

transcript: randomTranscript,

confidence: 0.95,

language: language,

duration: Math.random() \* 300 + 60 // Random duration between 1-6 minutes

});

}, 3000); // Simulate 3 second processing time

});

}

}

const transcriptionService = new WebTranscriptionService();

document.getElementById('transcriptionForm').addEventListener('submit', async (e) => {

e.preventDefault();

const videoUrl = document.getElementById('videoUrl').value;

const language = document.getElementById('language').value;

const fileInput = document.getElementById('audioFile');

const file = fileInput.files[0];

if (!videoUrl && !file) {

alert('Please enter a video URL or upload a file');

return;

}

// Show loading state

showLoading();

try {

// For demo purposes, using simulated transcription

const result = await transcriptionService.simulateTranscription(videoUrl, language, file);

if (result.success) {

showResult(result.transcript, videoUrl, language);

transcriptionService.currentTranscript = result.transcript;

transcriptionService.currentVideoUrl = videoUrl;

} else {

showError(result.error || 'Transcription failed');

}

} catch (error) {

showError(error.message);

} finally {

hideLoading();

}

});

function showLoading() {

document.getElementById('loading').style.display = 'block';

document.getElementById('result').style.display = 'none';

document.getElementById('submitBtn').disabled = true;

// Simulate progress

const progressBar = document.getElementById('progressBar');

const progressFill = document.getElementById('progressFill');

progressBar.style.display = 'block';

let progress = 0;

const interval = setInterval(() => {

progress += Math.random() \* 20;

if (progress > 90) progress = 90;

progressFill.style.width = progress + '%';

if (progress >= 90) {

clearInterval(interval);

}

}, 200);

}

function hideLoading() {

document.getElementById('loading').style.display = 'none';

document.getElementById('submitBtn').disabled = false;

document.getElementById('progressBar').style.display = 'none';

document.getElementById('progressFill').style.width = '100%';

}

function showResult(transcript, videoUrl, language) {

const resultDiv = document.getElementById('result');

const titleDiv = document.getElementById('resultTitle');

const transcriptDiv = document.getElementById('transcript');

titleDiv.textContent = '✅ Transcription Complete';

transcriptDiv.textContent = transcript;

resultDiv.className = 'result success';

resultDiv.style.display = 'block';

// Smooth scroll to result

resultDiv.scrollIntoView({ behavior: 'smooth' });

}

function showError(error) {

const resultDiv = document.getElementById('result');

const titleDiv = document.getElementById('resultTitle');

const transcriptDiv = document.getElementById('transcript');

titleDiv.textContent = '❌ Transcription Failed';

transcriptDiv.textContent = error;

resultDiv.className = 'result error';

resultDiv.style.display = 'block';

}

function copyTranscript() {

const transcript = transcriptionService.currentTranscript;

navigator.clipboard.writeText(transcript).then(() => {

alert('Transcript copied to clipboard!');

});

}

function downloadTranscript() {

const transcript = transcriptionService.currentTranscript;

const videoUrl = transcriptionService.currentVideoUrl;

const timestamp = new Date().toISOString().slice(0, 19).replace(/:/g, '-');

const content = `Video Transcription

URL: ${videoUrl}

Date: ${new Date().toLocaleString()}

Language: ${document.getElementById('language').value}

${'-'.repeat(50)}

${transcript}`;

const blob = new Blob([content], { type: 'text/plain' });

const url = URL.createObjectURL(blob);

const a = document.createElement('a');

a.href = url;

a.download = `transcript\_${timestamp}.txt`;

document.body.appendChild(a);

a.click();

document.body.removeChild(a);

URL.revokeObjectURL(url);

}

function shareTranscript() {

if (navigator.share) {

navigator.share({

title: 'Video Transcript',

text: transcriptionService.currentTranscript,

url: transcriptionService.currentVideoUrl

});

} else {

// Fallback: copy link to clipboard

const shareUrl = `${window.location.origin}?transcript=${encodeURIComponent(transcriptionService.currentTranscript)}`;

navigator.clipboard.writeText(shareUrl).then(() => {

alert('Share link copied to clipboard!');

});

}

}

// Handle file upload display

document.getElementById('audioFile').addEventListener('change', function(e) {

const fileLabel = document.querySelector('.file-label');

if (e.target.files[0]) {

fileLabel.textContent = `📁 Selected: ${e.target.files[0].name}`;

document.getElementById('videoUrl').required = false;

} else {

fileLabel.textContent = '📁 Or upload audio/video file (optional)';

document.getElementById('videoUrl').required = true;

}

});

// URL input handling

document.getElementById('videoUrl').addEventListener('input', function(e) {

const fileInput = document.getElementById('audioFile');

if (e.target.value) {

fileInput.required = false;

} else if (!fileInput.files[0]) {

fileInput.required = true;

}

});

</script>

</body>

</html>