<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Code Cracker - Multi-Device Cryptography Tool</title>

<style>

:root {

--primary: #2c3e50;

--secondary: #3498db;

--accent: #e74c3c;

--light: #ecf0f1;

--dark: #2c3e50;

--success: #2ecc71;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

body {

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

color: var(--light);

min-height: 100vh;

padding: 20px;

display: flex;

flex-direction: column;

align-items: center;

}

.container {

width: 100%;

max-width: 800px;

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(10px);

border-radius: 15px;

padding: 25px;

box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);

margin: 20px 0;

}

header {

text-align: center;

margin-bottom: 30px;

}

h1 {

font-size: 2.5rem;

margin-bottom: 10px;

color: white;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);

}

.subtitle {

font-size: 1.2rem;

color: var(--light);

margin-bottom: 20px;

}

.input-group {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 8px;

font-weight: 600;

}

textarea, select, input {

width: 100%;

padding: 12px 15px;

border: none;

border-radius: 8px;

background: rgba(255, 255, 255, 0.9);

font-size: 1rem;

margin-bottom: 10px;

transition: all 0.3s ease;

}

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

outline: none;

box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.5);

}

textarea {

min-height: 120px;

resize: vertical;

}

.shift-container {

display: flex;

align-items: center;

gap: 15px;

margin: 15px 0;

}

input[type="range"] {

flex: 1;

}

.shift-value {

font-weight: bold;

font-size: 1.2rem;

min-width: 30px;

text-align: center;

}

.btn-group {

display: flex;

gap: 15px;

margin: 20px 0;

}

button {

flex: 1;

padding: 14px;

border: none;

border-radius: 8px;

font-weight: 600;

font-size: 1rem;

cursor: pointer;

transition: all 0.3s ease;

}

.encode-btn {

background: var(--secondary);

color: white;

}

.decode-btn {

background: var(--accent);

color: white;

}

.reset-btn {

background: var(--dark);

color: white;

}

button:hover {

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);

}

.result-container {

background: rgba(255, 255, 255, 0.9);

border-radius: 8px;

padding: 15px;

color: var(--dark);

margin-top: 20px;

}

.result-title {

font-weight: 600;

margin-bottom: 10px;

color: var(--primary);

}

#result {

min-height: 100px;

line-height: 1.6;

}

.instructions {

margin-top: 30px;

padding: 20px;

background: rgba(255, 255, 255, 0.1);

border-radius: 10px;

}

.instructions h2 {

margin-bottom: 15px;

color: white;

}

.instructions ol {

padding-left: 20px;

line-height: 1.8;

}

.instructions li {

margin-bottom: 10px;

}

.download-section {

text-align: center;

margin-top: 30px;

padding: 20px;

}

.download-btn {

display: inline-block;

padding: 12px 30px;

background: var(--success);

color: white;

text-decoration: none;

border-radius: 8px;

font-weight: 600;

transition: all 0.3s ease;

}

.download-btn:hover {

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);

}

footer {

text-align: center;

margin-top: 40px;

padding: 20px;

color: var(--light);

font-size: 0.9rem;

}

@media (max-width: 600px) {

.btn-group {

flex-direction: column;

}

h1 {

font-size: 2rem;

}

.container {

padding: 15px;

}

}

</style>

</head>

<body>

<div class="container">

<header>

<h1>🔐 Code Cracker</h1>

<p class="subtitle">Encode and decode messages with multiple cipher algorithms</p>

</header>

<div class="input-group">

<label for="message">Your Message:</label>

<textarea id="message" placeholder="Enter text to encode or decode..."></textarea>

</div>

<div class="input-group">

<label for="cipher">Select Cipher:</label>

<select id="cipher">

<option value="caesar">Caesar Cipher</option>

<option value="atbash">Atbash Cipher</option>

<option value="reverse">Reverse Text</option>

<option value="rot13">ROT13</option>

</select>

</div>

<div class="input-group">

<label for="shift">Cipher Shift (Key):</label>

<div class="shift-container">

<input type="range" id="shift" min="1" max="25" value="3">

<span id="shiftValue" class="shift-value">3</span>

</div>

</div>

<div class="btn-group">

<button class="encode-btn" onclick="encode()">Encode</button>

<button class="decode-btn" onclick="decode()">Decode</button>

<button class="reset-btn" onclick="resetForm()">Reset</button>

</div>

<div class="result-container">

<div class="result-title">Result:</div>

<div id="result"></div>

</div>

</div>

<div class="container instructions">

<h2>How to Use Code Cracker</h2>

<ol>

<li>Type your message in the text area</li>

<li>Select a cipher algorithm from the dropdown</li>

<li>Adjust the shift value if using Caesar cipher</li>

<li>Click "Encode" to encrypt or "Decode" to decrypt</li>

<li>Use the "Reset" button to clear everything</li>

</ol>

<h2>About the Ciphers</h2>

<ul>

<li><strong>Caesar Cipher:</strong> Shifts each letter by a fixed number down the alphabet</li>

<li><strong>Atbash Cipher:</strong> Replaces each letter with its mirror (A=Z, B=Y, etc.)</li>

<li><strong>Reverse Text:</strong> Reverses the order of all characters</li>

<li><strong>ROT13:</strong> Special case of Caesar cipher with a shift of 13</li>

</ul>

</div>

<div class="download-section">

<a href="#" class="download-btn" onclick="downloadCode()">Download Code Files</a>

<p style="margin-top: 15px;">Save this project to your computer and upload to GitHub</p>

</div>

<footer>

<p>Code Cracker Web Application &copy; 2023 | Works on all devices</p>

</footer>

<script>

// DOM elements

const shiftSlider = document.getElementById('shift');

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

const messageInput = document.getElementById('message');

const cipherSelect = document.getElementById('cipher');

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

// Update shift value display when slider moves

shiftSlider.addEventListener('input', () => {

shiftValue.textContent = shiftSlider.value;

});

// Caesar Cipher algorithm

function caesarCipher(text, shift, decode = false) {

shift = decode ? (26 - shift) % 26 : shift;

return text.replace(/[a-z]/gi, (letter) => {

const shiftBase = letter < 'a' ? 65 : 97;

const shifted = (letter.charCodeAt(0) - shiftBase + shift) % 26;

const finalCode = shifted >= 0 ? shifted + shiftBase : shifted + shiftBase + 26;

return String.fromCharCode(finalCode);

});

}

// Atbash Cipher algorithm

function atbashCipher(text) {

return text.replace(/[a-z]/gi, (letter) => {

const code = letter.charCodeAt(0);

const isUpper = code >= 65 && code <= 90;

const base = isUpper ? 65 : 97;

return String.fromCharCode(25 - (code - base) + base);

});

}

// Reverse Text algorithm

function reverseText(text) {

return text.split('').reverse().join('');

}

// ROT13 algorithm (special case of Caesar)

function rot13Cipher(text) {

return caesarCipher(text, 13);

}

// Encode function

function encode() {

const message = messageInput.value;

const cipherType = cipherSelect.value;

const shift = parseInt(shiftSlider.value);

let result = '';

if (!message) {

resultDiv.innerHTML = '<span style="color: #e74c3c;">Please enter a message first!</span>';

return;

}

switch(cipherType) {

case 'caesar':

result = caesarCipher(message, shift);

break;

case 'atbash':

result = atbashCipher(message);

break;

case 'reverse':

result = reverseText(message);

break;

case 'rot13':

result = rot13Cipher(message);

break;

}

resultDiv.textContent = result;

}

// Decode function

function decode() {

const message = messageInput.value;

const cipherType = cipherSelect.value;

const shift = parseInt(shiftSlider.value);

let result = '';

if (!message) {

resultDiv.innerHTML = '<span style="color: #e74c3c;">Please enter a message first!</span>';

return;

}

switch(cipherType) {

case 'caesar':

result = caesarCipher(message, shift, true);

break;

case 'atbash':

// Atbash is self-reversible (encode and decode are the same)

result = atbashCipher(message);

break;

case 'reverse':

// Reverse is self-reversible

result = reverseText(message);

break;

case 'rot13':

// ROT13 is self-reversible

result = rot13Cipher(message);

break;

}

resultDiv.textContent = result;

}

// Reset form function

function resetForm() {

messageInput.value = '';

shiftSlider.value = 3;

shiftValue.textContent = '3';

cipherSelect.value = 'caesar';

resultDiv.textContent = '';

}

// Download code function

function downloadCode() {

// Create a zip file with all the code (simulated here with instructions)

alert('To download the code:\n1. Right-click on the page\n2. Select "Save As"\n3. Save the file as "code-cracker.html"\n\nYou can then upload this HTML file to GitHub!');

}

</script>

</body>

</html>