<!DOCTYPE html>

<html lang=*"en"*>

<head>

    <meta charset=*"UTF-8"*>

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

    <title>Brute-Force Attack Simulation</title>

    <style>

        \* {

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

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

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

            min-height: 100vh;

            padding: 20px;

        }

        .container {

            max-width: 1000px;

            margin: 0 auto;

            background: white;

            border-radius: 15px;

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

            overflow: hidden;

        }

        .header {

            background: linear-gradient(135deg, #2c3e50 0%, #3498db 100%);

            color: white;

            padding: 30px;

            text-align: center;

        }

        .content {

            padding: 30px;

        }

        .section {

            background: #f8f9fa;

            margin: 20px 0;

            padding: 25px;

            border-radius: 10px;

            border-left: 5px solid #3498db;

        }

        .section h3 {

            color: #2c3e50;

            margin-bottom: 15px;

            font-size: 1.3rem;

        }

        .input-group {

            margin: 15px 0;

        }

        .input-group label {

            display: block;

            margin-bottom: 8px;

            font-weight: 600;

            color: #2c3e50;

        }

        .input-group input, .input-group textarea {

            width: 100%;

            padding: 12px;

            border: 2px solid #ddd;

            border-radius: 8px;

            font-size: 16px;

            font-family: *'Courier New'*, monospace;

        }

        .input-group input:focus, .input-group textarea:focus {

            outline: none;

            border-color: #3498db;

        }

        .btn {

            background: linear-gradient(135deg, #3498db 0%, #2980b9 100%);

            color: white;

            border: none;

            padding: 12px 25px;

            border-radius: 8px;

            cursor: pointer;

            font-size: 16px;

            margin: 5px;

            transition: transform 0.2s ease;

        }

        .btn:hover {

            transform: translateY(-2px);

        }

        .btn.danger {

            background: linear-gradient(135deg, #e74c3c 0%, #c0392b 100%);

        }

        .btn.success {

            background: linear-gradient(135deg, #27ae60 0%, #229954 100%);

        }

        .results {

            background: white;

            border-radius: 8px;

            padding: 15px;

            margin-top: 15px;

            max-height: 300px;

            overflow-y: auto;

            border: 2px solid #ecf0f1;

        }

        .attempt {

            padding: 10px;

            margin: 8px 0;

            border-radius: 5px;

            border-left: 4px solid #e74c3c;

            background: #fff5f5;

        }

        .attempt.success {

            border-left-color: #27ae60;

            background: #f0fff4;

        }

        .explanation {

            background: #e8f4fd;

            padding: 20px;

            border-radius: 8px;

            margin: 20px 0;

            border-left: 4px solid #3498db;

        }

        .explanation h4 {

            color: #2980b9;

            margin-bottom: 10px;

        }

        .stats {

            display: flex;

            justify-content: space-around;

            margin: 20px 0;

            flex-wrap: wrap;

        }

        .stat {

            text-align: center;

            padding: 15px;

            background: #3498db;

            color: white;

            border-radius: 8px;

            margin: 5px;

            min-width: 120px;

        }

        .stat-value {

            font-size: 1.5rem;

            font-weight: bold;

        }

        .progress {

            width: 100%;

            height: 20px;

            background: #ecf0f1;

            border-radius: 10px;

            overflow: hidden;

            margin: 10px 0;

        }

        .progress-bar {

            height: 100%;

            background: linear-gradient(90deg, #3498db, #2980b9);

            width: 0%;

            transition: width 0.3s ease;

        }

        @media (max-width: 768px) {

            .stats {

                flex-direction: column;

            }

            .content {

                padding: 15px;

            }

        }

    </style>

</head>

<body>

    <div class=*"container"*>

        <div class=*"header"*>

            <h1>🔐 Caesar Cipher Brute-Force Simulator</h1>

            <p>Learn how weak encryption can be easily broken</p>

        </div>

        <div class=*"content"*>

*<!-- How it works explanation -->*

            <div class=*"explanation"*>

                <h4>🎯 How This Works:</h4>

                <p><strong>Step 1:</strong> Enter your plain text → We encrypt it using Caesar cipher<br>

                <strong>Step 2:</strong> Our "attacker" tries all 26 possible keys (brute-force)<br>

                <strong>Step 3:</strong> Each attempt is checked to see if it looks like English<br>

                <strong>Step 4:</strong> Successful decryptions are highlighted in green!</p>

            </div>

*<!-- Step 1: Create Encrypted Message -->*

            <div class=*"section"*>

                <h3>📝 Step 1: Create Your Message</h3>

                <div class=*"input-group"*>

                    <label>Enter Plain Text:</label>

                    <textarea id=*"plainText"* rows=*"2"* placeholder=*"Enter your message (e.g., HELLO WORLD)"*>HELLO WORLD</textarea>

                </div>

                <div class=*"input-group"*>

                    <label>Choose Encryption Key (1-25):</label>

                    <input type=*"number"* id=*"encryptKey"* min=*"1"* max=*"25"* value=*"3"*>

                </div>

                <button class=*"btn success"* onclick=*"encryptMessage()"*>🔒 Encrypt Message</button>

                <div id=*"encryptedResult"* style=*"display:none;"*>

                    <h4 style=*"margin-top: 15px;"*>Encrypted Message:</h4>

                    <div style=*"background:#2c3e50; color:white; padding:15px; border-radius:8px; font-family:'Courier New';"*>

                        <span id=*"encryptedText"*></span>

                    </div>

                </div>

            </div>

*<!-- Step 2: Brute Force Attack -->*

            <div class=*"section"*>

                <h3>⚔️ Step 2: Launch Brute-Force Attack</h3>

                <p>Now our "attacker" will try to crack your encrypted message by testing all possible keys!</p>

                <div class=*"input-group"*>

                    <label>Attack Speed:</label>

                    <input type=*"range"* id=*"speed"* min=*"200"* max=*"2000"* value=*"800"* oninput=*"updateSpeed()"*>

                    <span id=*"speedDisplay"*>800ms per attempt</span>

                </div>

                <button class=*"btn"* id=*"startBtn"* onclick=*"startAttack()"*>🚀 Start Attack</button>

                <button class=*"btn danger"* id=*"stopBtn"* onclick=*"stopAttack()"* style=*"display:none;"*>⏹️ Stop</button>

            </div>

*<!-- Progress and Stats -->*

            <div class=*"section"* id=*"progressSection"* style=*"display:none;"*>

                <h3>📊 Attack Progress</h3>

                <div class=*"progress"*>

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

                </div>

                <div id=*"currentStatus"*>Ready to attack...</div>

                <div class=*"stats"*>

                    <div class=*"stat"*>

                        <div class=*"stat-value"* id=*"attempts"*>0</div>

                        <div>Attempts</div>

                    </div>

                    <div class=*"stat"*>

                        <div class=*"stat-value"* id=*"timeElapsed"*>0s</div>

                        <div>Time</div>

                    </div>

                    <div class=*"stat"*>

                        <div class=*"stat-value"* id=*"cracked"*>0</div>

                        <div>Found</div>

                    </div>

                </div>

            </div>

*<!-- Results -->*

            <div class=*"section"* id=*"resultsSection"* style=*"display:none;"*>

                <h3>🔍 Attack Results</h3>

                <div class=*"results"* id=*"results"*></div>

            </div>

        </div>

    </div>

    <script>

*// Global variables*

        let attackInterval;

        let startTime;

        let currentTry = 0;

        let attempts = 0;

        let crackedCount = 0;

        let targetMessage = *''*;

        let originalMessage = *''*;

*// Caesar cipher functions*

        function caesarEncrypt(*text*, *shift*) {

            return text.split(*''*).map(*char* => {

                if (char.match(*/[A-Z]/*)) {

                    return String.fromCharCode(((char.charCodeAt(0) - 65 + shift) % 26) + 65);

                } else if (char.match(*/[a-z]/*)) {

                    return String.fromCharCode(((char.charCodeAt(0) - 97 + shift) % 26) + 97);

                }

                return char;

            }).join(*''*);

        }

        function caesarDecrypt(*text*, *shift*) {

            return caesarEncrypt(text, 26 - shift);

        }

*// Check if decrypted text matches original message*

        function isCorrectDecryption(*decryptedText*, *originalText*) {

            return decryptedText.toUpperCase() === originalText.toUpperCase();

        }

*// Check if text looks like English (for educational purpose)*

        function looksLikeEnglish(*text*) {

            const common = [*'THE'*, *'AND'*, *'FOR'*, *'ARE'*, *'BUT'*, *'NOT'*, *'YOU'*, *'ALL'*, *'CAN'*, *'HER'*, *'WAS'*, *'ONE'*, *'OUR'*, *'OUT'*, *'DAY'*, *'GET'*, *'HAS'*, *'HIM'*, *'HIS'*, *'HOW'*, *'ITS'*, *'MAY'*, *'NEW'*, *'NOW'*, *'OLD'*, *'SEE'*, *'TWO'*, *'WHO'*, *'BOY'*, *'DID'*];

            const words = text.toUpperCase().match(*/[A-Z]+/g*) || [];

            if (words.length === 0) return false;

            const matches = words.filter(*word* => common.includes(word)).length;

            return matches > 0 || words.some(*word* => word.length > 4);

        }

*// Update speed display*

        function updateSpeed() {

            const speed = document.getElementById(*'speed'*).value;

            document.getElementById(*'speedDisplay'*).textContent = speed + *'ms per attempt'*;

        }

*// Encrypt the message*

        function encryptMessage() {

            const plainText = document.getElementById(*'plainText'*).value.trim().toUpperCase();

            const key = parseInt(document.getElementById(*'encryptKey'*).value);

            if (!plainText) {

                alert(*'Please enter some text to encrypt!'*);

                return;

            }

            originalMessage = plainText; *// Store original for comparison*

            targetMessage = caesarEncrypt(plainText, key);

            document.getElementById(*'encryptedText'*).textContent = targetMessage;

            document.getElementById(*'encryptedResult'*).style.display = *'block'*;

        }

*// Start brute force attack*

        function startAttack() {

            if (!targetMessage) {

                alert(*'Please encrypt a message first!'*);

                return;

            }

*// Reset everything*

            currentTry = 1;

            attempts = 0;

            crackedCount = 0;

            startTime = Date.now();

*// Show progress sections*

            document.getElementById(*'progressSection'*).style.display = *'block'*;

            document.getElementById(*'resultsSection'*).style.display = *'block'*;

            document.getElementById(*'results'*).innerHTML = *''*;

*// Switch buttons*

            document.getElementById(*'startBtn'*).style.display = *'none'*;

            document.getElementById(*'stopBtn'*).style.display = *'inline-block'*;

*// Start the attack*

            const speed = parseInt(document.getElementById(*'speed'*).value);

            attackInterval = setInterval(() => {

                if (currentTry > 25) {

                    stopAttack();

                    return;

                }

                performAttack();

                currentTry++;

            }, speed);

        }

*// Perform one attack attempt*

        function performAttack() {

            attempts++;

            const decrypted = caesarDecrypt(targetMessage, currentTry);

            const isCorrect = isCorrectDecryption(decrypted, originalMessage);

            const looksEnglish = looksLikeEnglish(decrypted);

*// Only count as cracked if it's the correct decryption*

            if (isCorrect) crackedCount++;

*// Update progress*

            const progress = (currentTry / 25) \* 100;

            document.getElementById(*'progressBar'*).style.width = progress + *'%'*;

            document.getElementById(*'currentStatus'*).textContent =

*`Trying key ${currentTry}: ${decrypted}`*;

*// Update stats*

            const elapsed = Math.floor((Date.now() - startTime) / 1000);

            document.getElementById(*'attempts'*).textContent = attempts;

            document.getElementById(*'timeElapsed'*).textContent = elapsed + *'s'*;

            document.getElementById(*'cracked'*).textContent = crackedCount;

*// Add result - show correct match in green, possible English in yellow, others in red*

            const resultDiv = document.createElement(*'div'*);

            let statusText = *''*;

            let statusColor = *'#e74c3c'*;

            if (isCorrect) {

                resultDiv.className = *'attempt success'*;

                statusText = *'✅ CORRECT MATCH!'*;

                statusColor = *'#27ae60'*;

            } else if (looksEnglish) {

                resultDiv.className = *'attempt'*;

                resultDiv.style.borderLeftColor = *'#f39c12'*;

                resultDiv.style.backgroundColor = *'#fef9e7'*;

                statusText = *'⚠️ Looks like English'*;

                statusColor = *'#f39c12'*;

            } else {

                resultDiv.className = *'attempt'*;

                statusText = *'❌ Not English'*;

            }

            resultDiv.innerHTML = *`*

*<strong>Key ${currentTry}:</strong> ${decrypted}*

*<span style="color:${statusColor};">${statusText}</span>*

*`*;

            document.getElementById(*'results'*).appendChild(resultDiv);

            document.getElementById(*'results'*).scrollTop = document.getElementById(*'results'*).scrollHeight;

        }

*// Stop the attack*

        function stopAttack() {

            if (attackInterval) {

                clearInterval(attackInterval);

                attackInterval = null;

            }

            document.getElementById(*'startBtn'*).style.display = *'inline-block'*;

            document.getElementById(*'stopBtn'*).style.display = *'none'*;

            document.getElementById(*'currentStatus'*).textContent = *'Attack completed!'*;

        }

*// Initialize with example*

        window.onload = function() {

            encryptMessage();

        };

    </script>

</body>

</html>