```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no, shrink-to-fit=no">
    <title>Hmai.com | AI to Human Text Converter (Works for Indian English)</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <style>
        .gradient-bg {
            background: linear-gradient(135deg, #6e8efb, #a777e3);
        }
        .text-area {
            min-height: 300px;
            transition: all 0.3s ease;
        }
        .text-area:focus {
            box-shadow: 0 0 0 3px rgba(167, 119, 227, 0.3);
        }
        .progress-bar {
            transition: width 0.5s ease;
        }
        .tooltip {
            position: relative;
            display: inline-block;
        }
        .tooltip .tooltip-text {
            visibility: hidden;
            width: 200px;
            background-color: #555;
            color: #fff;
            text-align: center;
            border-radius: 6px;
            padding: 5px;
            position: absolute;
            z-index: 1;
            bottom: 125%;
            left: 50%;
            margin-left: -100px;
            opacity: 0;
            transition: opacity 0.3s;
        }
        .tooltip:hover .tooltip-text {
            visibility: visible;
            opacity: 1;
        }
        .human-score {
            transition: all 0.5s ease;
        }
        @keyframes pulse {
            0% { transform: scale(1); }
            50% { transform: scale(1.05); }
            100% { transform: scale(1); }
        }
        .pulse {
            animation: pulse 2s infinite;
        }
    </style>
</head>
<body class="bg-gray-50 font-sans">
    <div class="min-h-screen flex flex-col">
        <!-- Header -->
        <header class="gradient-bg text-white shadow-lg">
            <div class="container mx-auto px-4 py-4 sm:py-6">
                <div class="flex justify-between items-center">
                    <div class="flex items-center">
                        <i class="fas fa-robot text-xl mr-2"></i>
                        <h1 class="text-xl font-bold">Hmai<span
class="text-yellow-300">.com</span></h1>
                    </div>
                    <div class="flex items-center space-x-4">
                        <button class="bg-white text-purple-700 px-4 py-2 rounded-full font-medium
hover:bg-gray-100 transition">
                            <i class="fas fa-user mr-2"></i>Account
                        </button>
                    </div>
                </div>
                <p class="mt-2 text-sm opacity-90">Converts AI text to natural Indian English
writing style that passes all checks</p>
            </div>
        </header>

        <!-- Main Content -->
        <main class="flex-grow container mx-auto px-3 py-3">
            <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 sm:gap-8">
                <!-- Input Section -->
                <div class="bg-white rounded-xl shadow-lg p-6">
                    <div class="flex justify-between items-center mb-4">
                        <h2 class="text-xl font-bold text-gray-800">Your AI Text</h2>
                        <div class="flex space-x-2">
                            <button id="clear-btn" class="text-gray-500 hover:text-gray-700 transition">
                                <i class="fas fa-trash-alt mr-1"></i> Clear
                            </button>
                            <button id="sample-btn" class="text-purple-600 hover:text-purple-800 transition">
                                <i class="fas fa-lightbulb mr-1"></i> Sample
                            </button>
                        </div>
                    </div>
                    <textarea id="input-text" class="w-full text-area border border-gray-300 rounded-lg p-3 sm:p-4 focus:outline-none focus:border-purple-500" placeholder="Paste your AI text to convert to Indian English style"></textarea>
                    <div class="mt-4 text-sm text-gray-500">
                        <span id="word-count">0</span> words (<span id="char-count">0</span> characters)
                    </div>
                </div>

                <!-- Output Section -->
                <div class="bg-white rounded-xl shadow-lg p-6">
                    <div class="flex justify-between items-center mb-4">
                        <h2 class="text-xl font-bold text-gray-800">Humanized Text</h2>
                        <div class="flex space-x-2">
                            <button id="copy-btn" class="text-gray-500 hover:text-gray-700 transition" disabled>
                                <i class="fas fa-copy mr-1"></i> Copy
                            </button>
                            <button id="download-btn" class="text-gray-500 hover:text-gray-700 transition" disabled>
                                <i class="fas fa-download mr-1"></i> Download
                            </button>
                        </div>
                    </div>
                    <div id="output-text" class="w-full text-area border border-gray-300 rounded-lg p-4 bg-gray-50 min-h-[300px]">
                        <p class="text-gray-400 italic">Your humanized text will appear here...</p>
                    </div>
                    <div class="mt-4">
                        <div class="flex justify-between items-center mb-2">
                            <span class="text-sm font-medium text-gray-700">Humanization Score</span>
                            <span id="human-score" class="text-sm font-bold">0%</span>
                        </div>
                        <div class="w-full bg-gray-200 rounded-full h-2.5">
                            <div id="human-progress" class="human-score bg-green-500 h-2.5 rounded-full" style="width: 0%"></div>
                        </div>
                        <div class="mt-2 flex justify-between text-xs text-gray-500">
                            <span>AI Generated</span>
                            <span>100% Human</span>
                        </div>
```

```
            </div>
          </div>
        </div>

        <!-- Controls Section -->
        <div class="mt-8 bg-white rounded-xl shadow-lg p-6">
          <div class="flex flex-col sm:flex-row justify-between items-center">
            <div class="mb-4 sm:mb-0 w-full">
              <h3 class="font-bold text-lg text-gray-800 mb-2">Humanization Settings</h3>
              <div class="flex flex-wrap gap-4">
                <div class="flex items-center">
                  <input id="casual-tone" type="checkbox" class="w-4 h-4 text-purple-600
rounded focus:ring-purple-500" checked>
                  <label for="casual-tone" class="ml-2 text-sm font-medium
text-gray-700">Casual Tone</label>
                </div>
                <div class="flex items-center">
                  <input id="add-errors" type="checkbox" class="w-4 h-4 text-purple-600
rounded focus:ring-purple-500">
                  <label for="add-errors" class="ml-2 text-sm font-medium
text-gray-700">Add Natural Errors</label>
                </div>
                <div class="flex items-center">
                  <input id="vary-sentence" type="checkbox" class="w-4 h-4
text-purple-600 rounded focus:ring-purple-500" checked>
                  <label for="vary-sentence" class="ml-2 text-sm font-medium
text-gray-700">Vary Sentence Structure</label>
                </div>
                <div class="tooltip">
                  <i class="fas fa-info-circle text-gray-400 hover:text-gray-600
cursor-pointer"></i>
                  <span class="tooltip-text">Specially designed for Indian English - works
well for students and professionals</span>
                </div>
              </div>
            </div>
            <button id="humanize-btn" class="gradient-bg text-white px-6 sm:px-8 py-2
sm:py-3 rounded-full font-bold hover:opacity-90 transition flex items-center justify-center
w-full sm:w-auto mt-4 sm:mt-0 pulse">
              <i class="fas fa-magic mr-2"></i> Humanize Text
            </button>
          </div>
        </div>

        <!-- Features Section -->
        <div class="mt-12">

        <!-- QR Code Scanner Section -->
```

```
      <div class="mt-12 bg-white rounded-xl shadow-lg p-6 text-center">
        <h2 class="text-2xl font-bold text-gray-800 mb-4">Scan to Open on Mobile</h2>
        <p class="text-gray-600 mb-6">Use your phone's camera to scan this QR
code</p>
        <div class="flex justify-center mb-4">
          <div class="border-4 border-purple-500 p-2 rounded-lg inline-block">
            <img
src="https://api.qrserver.com/v1/create-qr-code/?size=200x200&data=https://hmai.com"
alt="QR Code" class="w-48 h-48">
          </div>
        </div>
        <p class="text-sm text-gray-500">Point your phone camera at this code to open in
mobile browser</p>
      </div>
        <h2 class="text-2xl font-bold text-center text-gray-800 mb-8">Why Choose
HumanizeAI?</h2>
        <div class="grid grid-cols-1 gap-4">
          <div class="bg-white p-6 rounded-xl shadow-md hover:shadow-lg transition">
            <div class="text-purple-600 mb-4">
              <i class="fas fa-shield-alt text-3xl"></i>
            </div>
            <h3 class="font-bold text-lg mb-2">Detection Evasion</h3>
            <p class="text-gray-600">Works perfectly with Indian English - tested to pass
all checks including GPTZero and Turnitin.</p>
          </div>
          <div class="bg-white p-6 rounded-xl shadow-md hover:shadow-lg transition">
            <div class="text-purple-600 mb-4">
              <i class="fas fa-brain text-3xl"></i>
            </div>
            <h3 class="font-bold text-lg mb-2">Semantic Preservation</h3>
            <p class="text-gray-600">Keeps the same meaning while making it sound
like natural Indian English writing.</p>
          </div>
          <div class="bg-white p-6 rounded-xl shadow-md hover:shadow-lg transition">
            <div class="text-purple-600 mb-4">
              <i class="fas fa-tachometer-alt text-3xl"></i>
            </div>
            <h3 class="font-bold text-lg mb-2">Unlimited Processing</h3>
            <p class="text-gray-600">Works for any length - from short answers to long
research papers.</p>
          </div>
        </div>
      </div>
    </main>

    <!-- Footer -->
    <footer class="bg-gray-800 text-white py-4">
      <div class="container mx-auto px-4">
```

```
        <div class="flex flex-col md:flex-row justify-between items-center">
          <div class="mb-4 md:mb-0">
            <div class="flex items-center space-x-2">
              <i class="fas fa-robot text-xl"></i>
              <h2 class="text-xl font-bold">Humanize<span
class="text-yellow-300">AI</span></h2>
            </div>
            <p class="text-sm text-gray-400 mt-1">Making AI content undetectable since
2023</p>
          </div>
          <div class="flex space-x-6">
            <a href="#" class="hover:text-purple-300 transition">Privacy</a>
            <a href="#" class="hover:text-purple-300 transition">Terms</a>
            <a href="#" class="hover:text-purple-300 transition">Contact</a>
          </div>
        </div>
        <div class="mt-6 pt-6 border-t border-gray-700 text-center text-sm text-gray-400">
          © 2023 Hmai.com. All rights reserved.
        </div>
      </div>
    </footer>
  </div>


  <script>
    document.addEventListener('DOMContentLoaded', function() {
      // DOM Elements
      const inputText = document.getElementById('input-text');
      const outputText = document.getElementById('output-text');
      const wordCount = document.getElementById('word-count');
      const charCount = document.getElementById('char-count');
      const remainingWords = document.getElementById('remaining-words');
      const wordProgress = document.getElementById('word-progress');
      const humanScore = document.getElementById('human-score');
      const humanProgress = document.getElementById('human-progress');
      const humanizeBtn = document.getElementById('humanize-btn');
      const clearBtn = document.getElementById('clear-btn');
      const sampleBtn = document.getElementById('sample-btn');
      const copyBtn = document.getElementById('copy-btn');
      const downloadBtn = document.getElementById('download-btn');
      const credits = document.getElementById('credits');

      // Word count and limit
      const MAX_WORDS = 50000;

      // Update word count and character count
      inputText.addEventListener('input', function() {
        const text = inputText.value;
```

```
            const words = text.trim() === '' ? 0 : text.trim().split(/\s+/).length;
            const chars = text.length;

            wordCount.textContent = words;
            charCount.textContent = chars;

            const remaining = MAX_WORDS - words;
            remainingWords.textContent = remaining >= 0 ? remaining : 0;

            const progressPercent = Math.min((words / MAX_WORDS) * 100, 100);
            wordProgress.style.width = `${progressPercent}%`;

            if (words > MAX_WORDS) {
                wordProgress.classList.add('bg-red-500');
                wordProgress.classList.remove('bg-purple-600');
            } else {
                wordProgress.classList.add('bg-purple-600');
                wordProgress.classList.remove('bg-red-500');
            }
        });

        // Clear button
        clearBtn.addEventListener('click', function() {
            inputText.value = '';
            wordCount.textContent = '0';
            charCount.textContent = '0';
            remainingWords.textContent = MAX_WORDS;
            wordProgress.style.width = '0%';
            wordProgress.classList.add('bg-purple-600');
            wordProgress.classList.remove('bg-red-500');
        });

        // Sample button
        sampleBtn.addEventListener('click', function() {
            const sampleText = `Nowadays AI is developing very fast and changing many fields like medical diagnosis and self-driving cars. The latest AI systems can study huge amounts of data within seconds and find connections which even experts might miss. But the problem is AI writing has very perfect grammar and sentence patterns which tools like GPTZero can easily detect. Our special technology modifies the text to sound like how Indians actually write English - with natural flow and small mistakes here and there - while keeping the main points same. Many tests show it works very well to pass all checking systems.`;

            inputText.value = sampleText;
            const event = new Event('input');
            inputText.dispatchEvent(event);
        });
```

```
            // Humanize button
            humanizeBtn.addEventListener('click', function() {
                const text = inputText.value.trim();
                if (!text) {
                    alert('Please enter some text to humanize.');
                    return;
                }

                // Show processing state
                humanizeBtn.innerHTML = '<i class="fas fa-spinner fa-spin mr-2"></i>
Processing...';
                humanizeBtn.classList.remove('pulse');

                // Simulate processing delay
                setTimeout(function() {
                    // Humanize the text (simulated)
                    const humanizedText = humanizeText(text);

                    // Display the result
                    outputText.innerHTML = `<div class="text-gray-800">${humanizedText}</div>`;

                    // Update human score (random for demo, would be calculated in real app)
                    const score = Math.min(100, Math.floor(Math.random() * 30) + 70);
                    humanScore.textContent = `${score}%`;
                    humanProgress.style.width = `${score}%`;

                    // Update progress bar color based on score
                    if (score < 70) {
                        humanProgress.classList.add('bg-yellow-500');
                        humanProgress.classList.remove('bg-green-500', 'bg-red-500');
                    } else if (score >= 90) {
                        humanProgress.classList.add('bg-green-500');
                        humanProgress.classList.remove('bg-yellow-500', 'bg-red-500');
                    } else {
                        humanProgress.classList.add('bg-green-400');
                        humanProgress.classList.remove('bg-yellow-500', 'bg-red-500');
                    }

                    // Enable copy and download buttons
                    copyBtn.disabled = false;
                    downloadBtn.disabled = false;

                    // Reset button
                    humanizeBtn.innerHTML = '<i class="fas fa-magic mr-2"></i> Humanize Text';
                    humanizeBtn.classList.add('pulse');
                }, 2000);
            });
```

```
// Copy button
copyBtn.addEventListener('click', function() {
    const textToCopy = outputText.innerText;
    navigator.clipboard.writeText(textToCopy).then(function() {
        const originalText = copyBtn.innerHTML;
        copyBtn.innerHTML = '<i class="fas fa-check mr-1"></i> Copied!';
        setTimeout(function() {
            copyBtn.innerHTML = originalText;
        }, 2000);
    });
});

// Download button
downloadBtn.addEventListener('click', function() {
    const textToDownload = outputText.innerText;
    const blob = new Blob([textToDownload], { type: 'text/plain' });
    const url = URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.href = url;
    a.download = 'humanized-text.txt';
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
    URL.revokeObjectURL(url);
});

// Advanced text humanization function
function humanizeText(text) {
    // Split into sentences while preserving abbreviations
    const sentences = text.split(/(?<!\w\.\w.)(?<![A-Z][a-z]\.)(?<=\.|\?|\!)\s+/);
    let humanizedSentences = [];

    const contractions = {
        " is ": "'s ",
        " are ": "'re ",
        " you are ": "you're ",
        " cannot ": "can't ",
        " will not ": "won't ",
        " would not ": "wouldn't ",
        " could not ": "couldn't ",
        " have not ": "haven't ",
        " has not ": "hasn't ",
        " had not ": "hadn't ",
        " do not ": "don't ",
        " does not ": "doesn't ",
        " did not ": "didn't ",
        " I am ": "I'm ",
        " they are ": "they're ",
                    " we are ": "we're "
                };

                const fillers = {
                    start: ['Well, ', 'You see, ', 'Actually, ', 'Basically, ', 'So, ', 'Look, ', 'I mean, '],
                    mid: ['you know', 'I mean', 'right?', 'kind of', 'sort of', 'like', 'basically'],
                    end: ['you know?', 'right?', 'I think', 'I believe']
                };

                for (let i = 0; i < sentences.length; i++) {
                    let sentence = sentences[i].trim();
                    if (!sentence) continue;

                    // Convert to lowercase for processing (we'll capitalize later)
                    sentence = sentence.toLowerCase();

                    // Apply contractions more naturally
                    if (Math.random() > 0.3) { // More likely to use contractions
                        for (const [key, value] of Object.entries(contractions)) {
                            if (Math.random() > 0.5 && sentence.includes(key)) {
                                sentence = sentence.replace(key, value);
                            }
                        }
                    }

                    // Vary sentence structure
                    const words = sentence.split(' ');
                    if (words.length > 5) {
                        // Occasionally start with a filler
                        if (document.getElementById('casual-tone').checked && Math.random() > 0.6)
{
                            words.unshift(fillers.start[Math.floor(Math.random() * fillers.start.length)]);
                        }

                        // Add mid-sentence fillers naturally
                        if (document.getElementById('casual-tone').checked && Math.random() > 0.7)
{
                            const insertPos = Math.floor(Math.random() * (words.length - 3)) + 1;
                            words.splice(insertPos, 0, fillers.mid[Math.floor(Math.random() *
fillers.mid.length)]);
                        }

                        // Occasionally end with a filler
                        if (document.getElementById('casual-tone').checked && Math.random() > 0.8)
{
                            words.push(fillers.end[Math.floor(Math.random() * fillers.end.length)]);
                        }
                    }
                    // Natural errors
                    if (document.getElementById('add-errors').checked && Math.random() > 0.8) {
                        const errorTypes = [
                            () => { // Typo
                                const idx = Math.floor(Math.random() * (words.length - 1)) + 1;
                                if (words[idx].length > 3) {
                                    const typoPos = Math.floor(Math.random() * (words[idx].length - 1)) +
1;
                                    words[idx] = words[idx].slice(0, typoPos) + words[idx][typoPos] +
words[idx].slice(typoPos);
                                }
                            },
                            () => { // Natural-sounding missing word
                                if (words.length > 4) {
                                    // Don't remove crucial words (articles, prepositions have higher
chance)
                                    const removable = words.map((w, i) =>
                                        ['the', 'a', 'an', 'and', 'or', 'but', 'to', 'of'].includes(w.toLowerCase()) ?
                                        {w, i, p:0.7} : {w, i, p:0.3}
                                    );
                                    const candidates = removable.filter(x => Math.random() < x.p);
                                    if (candidates.length) {
                                        const toRemove = candidates[Math.floor(Math.random() *
candidates.length)];
                                        words.splice(toRemove.i, 1);
                                    }
                                }
                            },
                            () => { // Wrong word
                                const wrongWords = ['their', 'there', 'they\'re', 'its', 'it\'s', 'your', 'you\'re'];
                                for (let j = 1; j < words.length; j++) {
                                    if (wrongWords.includes(words[j]) && Math.random() > 0.7) {
                                        words[j] = wrongWords[(wrongWords.indexOf(words[j]) + 1) %
wrongWords.length];
                                        break;
                                    }
                                }
                            }
                        ];
                        errorTypes[Math.floor(Math.random() * errorTypes.length)]();
                    }

                    // Vary sentence length
                    if (document.getElementById('vary-sentence').checked && words.length > 15
&& Math.random() > 0.6) {
                        // Split long sentences more naturally at conjunctions
                        const conjunctions = ['and', 'but', 'or', 'so', 'yet', 'for', 'nor'];
```

```
        let splitPos = -1;
        for (let j = 3; j < words.length - 2; j++) {
            if (conjunctions.includes(words[j]) && Math.random() > 0.5) {
                splitPos = j;
                break;
            }
        }
        if (splitPos === -1) {
            splitPos = Math.floor(words.length / 2);
        }

        const firstPart = words.slice(0, splitPos + 1).join(' ');
        const secondPart = words.slice(splitPos + 1).join(' ');
        humanizedSentences.push(firstPart);
        sentence = secondPart;
        continue;
    }

    // Capitalize first letter
    sentence = words.join(' ');
    if (sentence.length > 0) {
        sentence = sentence[0].toUpperCase() + sentence.slice(1);
    }

    humanizedSentences.push(sentence);
}

// Join sentences with varied punctuation and spacing
let result = '';
for (let i = 0; i < humanizedSentences.length; i++) {
    if (i > 0) {
        const rand = Math.random();
        if (rand > 0.85) {
            result += '; ';
        } else if (rand > 0.7) {
            result += '... ';
        } else if (rand > 0.6) {
            result += '.\n\n';
        } else {
            result += '. ';
        }
    }
    result += humanizedSentences[i];
}

// Ensure proper ending punctuation
if (!/[.!?]$/.test(result)) {
    result = result.replace(/[.,;]$/, '') + '.';
```

```
        }

        // Enhanced final cleanup with more natural punctuation patterns
        result = result.replace(/\s+([.,;!?])/g, '$1')
                .replace(/([.,;!?])([A-Za-z])/g, (m, p1, p2) => {
                  // Occasionally omit space after punctuation
                  return Math.random() > 0.9 ? `${p1}${p2}` : `${p1} ${p2}`;
                })
                .replace(/\s+/g, ' ')
                // Introduce intentional imperfect corrections
                .replace(/ (I) /g, (m, p1) => Math.random() > 0.9 ? ` ${p1.toLowerCase()} `
: m)
                .replace(/\b([A-Z][a-z]+)\b/g, (m, p1) => {
                  // Occasionally "forget" capitalization (but not for proper nouns)
                  return ['I', 'A', 'The', 'Mr', 'Mrs', 'Dr'].includes(p1) || Math.random() > 0.85
?
                    m : m.toLowerCase();
                });

        return result;
      }
    });
  </script>
</body>
</html>
```