I'll help you build this AI text rewriter website. Let me create a complete implementation with all the features shown in your screenshots.

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>AI Text Rewriter & Converter</title>

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">

<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;

color: #333;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

.header {

text-align: center;

margin-bottom: 30px;

color: white;

}

.header h1 {

font-size: 2.5rem;

margin-bottom: 10px;

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

}

.header p {

font-size: 1.1rem;

opacity: 0.9;

}

.main-card {

background: white;

border-radius: 20px;

padding: 30px;

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

margin-bottom: 20px;

}

.input-section {

margin-bottom: 30px;

}

.input-label {

font-weight: 600;

margin-bottom: 10px;

display: block;

color: #555;

}

.text-input {

width: 100%;

min-height: 150px;

padding: 15px;

border: 2px solid #e1e5e9;

border-radius: 12px;

font-size: 16px;

resize: vertical;

transition: border-color 0.3s ease;

}

.text-input:focus {

outline: none;

border-color: #667eea;

}

.options-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 20px;

margin-bottom: 30px;

}

.option-group {

background: #f8f9fa;

padding: 20px;

border-radius: 12px;

border: 2px solid transparent;

transition: all 0.3s ease;

}

.option-group:hover {

border-color: #667eea;

transform: translateY(-2px);

}

.option-title {

font-weight: 600;

margin-bottom: 15px;

color: #333;

display: flex;

align-items: center;

gap: 8px;

}

.style-buttons, .converter-buttons {

display: flex;

flex-wrap: wrap;

gap: 8px;

}

.style-btn, .converter-btn {

padding: 8px 16px;

border: 2px solid #e1e5e9;

background: white;

border-radius: 20px;

cursor: pointer;

transition: all 0.3s ease;

font-size: 14px;

}

.style-btn:hover, .converter-btn:hover {

background: #667eea;

color: white;

border-color: #667eea;

}

.style-btn.active, .converter-btn.active {

background: #667eea;

color: white;

border-color: #667eea;

}

.action-buttons {

display: flex;

gap: 15px;

margin-bottom: 30px;

flex-wrap: wrap;

}

.btn {

padding: 12px 24px;

border: none;

border-radius: 25px;

cursor: pointer;

font-weight: 600;

transition: all 0.3s ease;

display: flex;

align-items: center;

gap: 8px;

}

.btn-primary {

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

color: white;

}

.btn-secondary {

background: #6c757d;

color: white;

}

.btn-success {

background: #28a745;

color: white;

}

.btn:hover {

transform: translateY(-2px);

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

}

.output-section {

margin-top: 30px;

}

.output-text {

width: 100%;

min-height: 150px;

padding: 15px;

border: 2px solid #e1e5e9;

border-radius: 12px;

font-size: 16px;

background: #f8f9fa;

resize: vertical;

}

.premium-banner {

background: linear-gradient(135deg, #ff6b6b 0%, #ee5a24 100%);

color: white;

padding: 20px;

border-radius: 15px;

margin-bottom: 20px;

text-align: center;

position: relative;

overflow: hidden;

}

.premium-banner::before {

content: '';

position: absolute;

top: -50%;

left: -50%;

width: 200%;

height: 200%;

background: repeating-linear-gradient(

45deg,

transparent,

transparent 10px,

rgba(255,255,255,0.1) 10px,

rgba(255,255,255,0.1) 20px

);

animation: shimmer 3s linear infinite;

}

@keyframes shimmer {

0% { transform: translateX(-100%); }

100% { transform: translateX(100%); }

}

.premium-content {

position: relative;

z-index: 1;

}

.premium-title {

font-size: 1.5rem;

margin-bottom: 10px;

font-weight: 700;

}

.premium-price {

font-size: 2rem;

font-weight: 700;

margin: 10px 0;

}

.premium-features {

display: flex;

justify-content: center;

gap: 20px;

margin: 15px 0;

flex-wrap: wrap;

}

.feature {

display: flex;

align-items: center;

gap: 5px;

font-size: 0.9rem;

}

.pricing-cards {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 20px;

margin-top: 30px;

}

.pricing-card {

background: white;

border-radius: 15px;

padding: 25px;

text-align: center;

border: 2px solid #e1e5e9;

transition: all 0.3s ease;

position: relative;

}

.pricing-card.featured {

border-color: #667eea;

transform: scale(1.05);

}

.pricing-card.featured::before {

content: 'BEST VALUE';

position: absolute;

top: -10px;

left: 50%;

transform: translateX(-50%);

background: #667eea;

color: white;

padding: 5px 15px;

border-radius: 15px;

font-size: 0.8rem;

font-weight: 600;

}

.plan-name {

font-size: 1.3rem;

font-weight: 700;

margin-bottom: 10px;

color: #333;

}

.plan-price {

font-size: 2.5rem;

font-weight: 700;

color: #667eea;

margin-bottom: 20px;

}

.plan-features {

list-style: none;

margin-bottom: 25px;

}

.plan-features li {

padding: 8px 0;

display: flex;

align-items: center;

gap: 10px;

}

.plan-features li::before {

content: '✓';

color: #28a745;

font-weight: bold;

}

.word-count {

font-size: 0.9rem;

color: #666;

margin-top: 10px;

}

.effects-section {

margin-top: 20px;

padding: 20px;

background: #f8f9fa;

border-radius: 12px;

}

.effect-controls {

display: grid;

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

gap: 15px;

margin-top: 15px;

}

.effect-control {

display: flex;

flex-direction: column;

gap: 8px;

}

.effect-control label {

font-weight: 600;

font-size: 0.9rem;

color: #555;

}

.effect-control select, .effect-control input {

padding: 8px 12px;

border: 2px solid #e1e5e9;

border-radius: 8px;

font-size: 14px;

}

.notification {

position: fixed;

top: 20px;

right: 20px;

padding: 15px 20px;

border-radius: 8px;

color: white;

font-weight: 600;

z-index: 1000;

transform: translateX(400px);

transition: transform 0.3s ease;

}

.notification.show {

transform: translateX(0);

}

.notification.success {

background: #28a745;

}

.notification.error {

background: #dc3545;

}

.loading {

display: none;

text-align: center;

padding: 20px;

}

.loading.show {

display: block;

}

.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 10px;

}

@keyframes spin {

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

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

}

@media (max-width: 768px) {

.container {

padding: 10px;

}

.header h1 {

font-size: 2rem;

}

.options-grid {

grid-template-columns: 1fr;

}

.action-buttons {

flex-direction: column;

}

.pricing-cards {

grid-template-columns: 1fr;

}

.premium-features {

flex-direction: column;

align-items: center;

}

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<h1><i class="fas fa-magic"></i> AI Text Rewriter & Converter</h1>

<p>Transform your text with AI-powered rewriting in multiple styles and creative text converters</p>

</div>

<!-- Premium Banner -->

<div class="premium-banner">

<div class="premium-content">

<div class="premium-title">🎉 LIMITED TIME OFFER</div>

<div class="premium-price">$0.99/month</div>

<div>or $10/year - Ends Oct 31</div>

<div class="premium-features">

<div class="feature"><i class="fas fa-check"></i> Unlimited AI rewrites</div>

<div class="feature"><i class="fas fa-check"></i> Unlimited conversions</div>

<div class="feature"><i class="fas fa-check"></i> Premium styles</div>

<div class="feature"><i class="fas fa-check"></i> No watermark</div>

</div>

<button class="btn btn-primary" onclick="upgradeToPremium()">

<i class="fas fa-crown"></i> Upgrade to Premium

</button>

</div>

</div>

<div class="main-card">

<!-- Input Section -->

<div class="input-section">

<label class="input-label">

<i class="fas fa-edit"></i> Your Text

</label>

<textarea

id="textInput"

class="text-input"

placeholder="Enter your text here to rewrite it in different styles..."

oninput="updateWordCount()"

></textarea>

<div class="word-count" id="wordCount">0 words</div>

</div>

<!-- Style Options -->

<div class="options-grid">

<div class="option-group">

<div class="option-title">

<i class="fas fa-palette"></i> Select Styles

</div>

<div class="style-buttons">

<button class="style-btn" onclick="toggleStyle(this, 'professional')">Professional</button>

<button class="style-btn" onclick="toggleStyle(this, 'funny')">Funny</button>

<button class="style-btn" onclick="toggleStyle(this, 'sarcastic')">Sarcastic</button>

<button class="style-btn" onclick="toggleStyle(this, 'persuasive')">Persuasive</button>

<button class="style-btn" onclick="toggleStyle(this, 'concise')">Concise</button>

<button class="style-btn" onclick="toggleStyle(this, 'formal')">Formal</button>

<button class="style-btn" onclick="toggleStyle(this, 'casual')">Casual</button>

<button class="style-btn" onclick="toggleStyle(this, 'creative')">Creative</button>

<button class="style-btn" onclick="toggleStyle(this, 'dramatic')">Dramatic</button>

<button class="style-btn" onclick="toggleStyle(this, 'storytelling')">Storytelling</button>

<button class="style-btn" onclick="toggleStyle(this, 'seo')">SEO</button>

<button class="style-btn" onclick="toggleStyle(this, 'boost')">BOOST IT!</button>

</div>

</div>

<div class="option-group">

<div class="option-title">

<i class="fas fa-tools"></i> Select Converter

</div>

<div class="converter-buttons">

<button class="converter-btn" onclick="toggleConverter(this, 'fancy')">Fancy Fonts</button>

<button class="converter-btn" onclick="toggleConverter(this, 'upside')">Upside Down</button>

<button class="converter-btn" onclick="toggleConverter(this, 'vertical')">Vertical</button>

<button class="converter-btn" onclick="toggleConverter(this, 'shadow')">Shadow Effects</button>

<button class="converter-btn" onclick="toggleConverter(this, 'neon')">Neon</button>

<button class="converter-btn" onclick="toggleConverter(this, 'backwards')">Backwards</button>

<button class="converter-btn" onclick="toggleConverter(this, 'wave')">Wave</button>

<button class="converter-btn" onclick="toggleConverter(this, 'glitch')">Glitch</button>

<button class="converter-btn" onclick="toggleConverter(this, 'emoji')">Emoji Wrapper</button>

</div>

</div>

</div>

<!-- Effects Section -->

<div class="effects-section">

<div class="option-title">

<i class="fas fa-sliders-h"></i> Effects & Customization

</div>

<div class="effect-controls">

<div class="effect-control">

<label>Shadow Type</label>

<select id="shadowType" onchange="applyEffects()">

<option value="none">None</option>

<option value="soft">Soft</option>

<option value="hard">Hard</option>

<option value="glow">Glow</option>

<option value="3d">3D</option>

</select>

</div>

<div class="effect-control">

<label>Color Palette</label>

<select id="colorPalette" onchange="applyEffects()">

<option value="default">Default</option>

<option value="rainbow">Rainbow</option>

<option value="neon">Neon</option>

<option value="pastel">Pastel</option>

</select>

</div>

<div class="effect-control">

<label>Font Size</label>

<input type="range" id="fontSize" min="12" max="24" value="16" onchange="applyEffects()">

</div>

</div>

</div>

<!-- Action Buttons -->

<div class="action-buttons">

<button class="btn btn-primary" onclick="rewriteText()">

<i class="fas fa-magic"></i> Rewrite Text

</button>

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

<i class="fas fa-exchange-alt"></i> Convert Text

</button>

<button class="btn btn-success" onclick="copyText()">

<i class="fas fa-copy"></i> Copy Text

</button>

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

<i class="fas fa-download"></i> Download as Image

</button>

</div>

<!-- Loading -->

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

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

<p>Processing your text...</p>

</div>

<!-- Output Section -->

<div class="output-section">

<label class="input-label">

<i class="fas fa-eye"></i> Preview

</label>

<div id="outputText" class="output-text" contenteditable="true"></div>

</div>

</div>

<!-- Pricing Section -->

<div class="main-card">

<h2 style="text-align: center; margin-bottom: 30px; color: #333;">

<i class="fas fa-crown"></i> Choose Your Premium Plan

</h2>

<div class="pricing-cards">

<div class="pricing-card">

<div class="plan-name">Monthly Plan</div>

<div class="plan-price">$5<span style="font-size: 1rem;">/month</span></div>

<ul class="plan-features">

<li>Unlimited AI rewrites</li>

<li>Unlimited text conversions</li>

<li>3 premium rewrite styles</li>

<li>5 premium converters</li>

<li>Premium shadows & colors</li>

<li>Dark mode</li>

<li>No watermark</li>

<li>Priority processing</li>

</ul>

<button class="btn btn-primary" style="width: 100%;" onclick="subscribeMonthly()">

Subscribe Monthly

</button>

</div>

<div class="pricing-card featured">

<div class="plan-name">Yearly Plan</div>

<div class="plan-price">$50<span style="font-size: 1rem;">/year</span></div>

<div style="color: #666; font-size: 0.9rem; margin-bottom: 20px;">Save $10 compared to monthly!</div>

<ul class="plan-features">

<li>Unlimited AI rewrites</li>

<li>Unlimited text conversions</li>

<li>3 premium rewrite styles</li>

<li>5 premium converters</li>

<li>Premium shadows & colors</li>

<li>Dark mode</li>

<li>No watermark</li>

<li>Priority processing</li>

</ul>

<button class="btn btn-primary" style="width: 100%;" onclick="subscribeYearly()">

Subscribe Yearly - BEST VALUE

</button>

</div>

</div>

</div>

</div>

<!-- Notification -->

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

<!-- Firebase SDK -->

<script src="https://www.gstatic.com/firebasejs/9.0.0/firebase-app-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.0.0/firebase-auth-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.0.0/firebase-firestore-compat.js"></script>

<script>

// Firebase Configuration

const firebaseConfig = {

apiKey: "AIzaSyDw7jPSr8uzu07Hd\_XtsGAyp7W6YJbBfTY",

authDomain: "text-rewrite-tool.firebaseapp.com",

projectId: "text-rewrite-tool",

storageBucket: "text-rewrite-tool.firebasestorage.app",

messagingSenderId: "904397412845",

appId: "1:904397412845:web:373b062e32bb69eb356fe4"

};

// Initialize Firebase

firebase.initializeApp(firebaseConfig);

const auth = firebase.auth();

const db = firebase.firestore();

// Global variables

let selectedStyles = [];

let selectedConverter = null;

let isPremium = false;

// OpenAI Configuration

const OPENAI\_API\_KEY = 'sk-proj-VWFtsje63VGIaZGJwy\_HnC5oAckyHOCu56kEgqnrLj1b75Lnoxp2xWmewwX7kABsK48Lku-n5RT3BlbkFJqlaHRZKoOwDnwQ5lFwUvJFMke0v\_K0ObkMfPG7AgW440yIvOWMEs5OngTDCxbAEjPkaAetm24A';

// Stripe Configuration

const stripe = Stripe('pk\_test\_51S7tP35pZ3kxpaj7dp5UYgAuiXdhNmXNaLJRxk0AIafKB7VcmSOxwNpZx8nGgxY4IBwatNjcIo45OJH0OCnLourj00fItjTnDk');

// Product IDs

const PRODUCT\_IDS = {

monthlyPromo: 'prod\_TFztGiAO5wwacH',

monthlyRegular: 'prod\_TG0MBOvJeovbxt',

yearlyPromo: 'prod\_TG1j9AzTdtR9RT',

yearlyRegular: 'prod\_TG1qE1Ta33BR6g'

};

// Initialize app

document.addEventListener('DOMContentLoaded', function() {

checkUserStatus();

loadUserPreferences();

});

// Check user authentication and premium status

function checkUserStatus() {

auth.onAuthStateChanged(async (user) => {

if (user) {

const userDoc = await db.collection('users').doc(user.uid).get();

if (userDoc.exists) {

isPremium = userDoc.data().isPremium || false;

updateUIBasedOnPremium();

}

}

});

}

// Update UI based on premium status

function updateUIBasedOnPremium() {

const premiumElements = document.querySelectorAll('.premium-feature');

premiumElements.forEach(el => {

if (isPremium) {

el.classList.remove('locked');

} else {

el.classList.add('locked');

}

});

}

// Toggle style selection

function toggleStyle(button, style) {

button.classList.toggle('active');

if (button.classList.contains('active')) {

selectedStyles.push(style);

} else {

selectedStyles = selectedStyles.filter(s => s !== style);

}

}

// Toggle converter selection

function toggleConverter(button, converter) {

document.querySelectorAll('.converter-btn').forEach(btn => {

btn.classList.remove('active');

});

button.classList.add('active');

selectedConverter = converter;

}

// Update word count

function updateWordCount() {

const text = document.getElementById('textInput').value;

const words = text.trim().split(/\s+/).filter(word => word.length > 0).length;

document.getElementById('wordCount').textContent = `${words} words`;

}

// Apply effects

function applyEffects() {

const output = document.getElementById('outputText');

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

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

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

output.style.fontSize = fontSize + 'px';

// Apply shadow effects

switch(shadowType) {

case 'soft':

output.style.textShadow = '2px 2px 4px rgba(0,0,0,0.3)';

break;

case 'hard':

output.style.textShadow = '3px 3px 0px #666';

break;

case 'glow':

output.style.textShadow = '0 0 10px #667eea';

break;

case '3d':

output.style.textShadow = '1px 1px 0px #ccc, 2px 2px 0px #ccc, 3px 3px 0px #ccc';

break;

default:

output.style.textShadow = 'none';

}

// Apply color palette

switch(colorPalette) {

case 'rainbow':

output.style.background = 'linear-gradient(45deg, #ff0000, #ff8000, #ffff00, #80ff00, #00ff00, #00ff80, #00ffff, #0080ff, #0000ff, #8000ff, #ff00ff, #ff0080)';

output.style.webkitBackgroundClip = 'text';

output.style.webkitTextFillColor = 'transparent';

break;

case 'neon':

output.style.color = '#00ff00';

output.style.textShadow = '0 0 5px #00ff00, 0 0 10px #00ff00, 0 0 15px #00ff00';

break;

case 'pastel':

output.style.color = '#ffb3ba';

break;

default:

output.style.color = '#333';

output.style.background = 'none';

output.style.webkitTextFillColor = 'initial';

}

}

// Rewrite text using OpenAI

async function rewriteText() {

const inputText = document.getElementById('textInput').value;

if (!inputText.trim()) {

showNotification('Please enter some text to rewrite', 'error');

return;

}

if (selectedStyles.length === 0) {

showNotification('Please select at least one style', 'error');

return;

}

if (!isPremium && selectedStyles.length > 1) {

showNotification('Free users can only select one style. Upgrade to premium for multiple styles!', 'error');

return;

}

showLoading(true);

try {

const response = await fetch('https://api.openai.com/v1/chat/completions', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

'Authorization': `Bearer ${OPENAI\_API\_KEY}`

},

body: JSON.stringify({

model: 'gpt-3.5-turbo',

messages: [{

role: 'user',

content: `Rewrite the following text in ${selectedStyles.join(', ')} style: "${inputText}"`

}],

max\_tokens: 1000,

temperature: 0.7

})

});

const data = await response.json();

const rewrittenText = data.choices[0].message.content;

document.getElementById('outputText').textContent = rewrittenText;

applyEffects();

showNotification('Text rewritten successfully!', 'success');

// Save to history

saveToHistory(inputText, rewrittenText, 'rewrite');

} catch (error) {

console.error('Error rewriting text:', error);

showNotification('Error rewriting text. Please try again.', 'error');

} finally {

showLoading(false);

}

}

// Convert text

function convertText() {

const inputText = document.getElementById('textInput').value;

if (!inputText.trim()) {

showNotification('Please enter some text to convert', 'error');

return;

}

if (!selectedConverter) {

showNotification('Please select a converter', 'error');

return;

}

if (!isPremium) {

showNotification('Text conversion is a premium feature. Please upgrade to use converters!', 'error');

return;

}

showLoading(true);

setTimeout(() => {

let convertedText = '';

switch(selectedConverter) {

case 'fancy':

convertedText = convertToFancyFonts(inputText);

break;

case 'upside':

convertedText = convertToUpsideDown(inputText);

break;

case 'vertical':

convertedText = convertToVertical(inputText);

break;

case 'shadow':

convertedText = convertToShadow(inputText);

break;

case 'neon':

convertedText = convertToNeon(inputText);

break;

case 'backwards':

convertedText = convertToBackwards(inputText);

break;

case 'wave':

convertedText = convertToWave(inputText);

break;

case 'glitch':

convertedText = convertToGlitch(inputText);

break;

case 'emoji':

convertedText = convertToEmojiWrapper(inputText);

break;

default:

convertedText = inputText;

}

document.getElementById('outputText').textContent = convertedText;

applyEffects();

showNotification('Text converted successfully!', 'success');

// Save to history

saveToHistory(inputText, convertedText, 'convert');

showLoading(false);

}, 1000);

}

// Text conversion functions

function convertToFancyFonts(text) {

const fancyFonts = {

'a': '𝓪', 'b': '𝓫', 'c': '𝓬', 'd': '𝓭', 'e': '𝓮', 'f': '𝓯', 'g': '𝓰', 'h': '𝓱', 'i': '𝓲', 'j': '𝓳',

'k': '𝓴', 'l': '𝓵', 'm': '𝓶', 'n': '𝓷', 'o': '𝓸', 'p': '𝓹', 'q': '𝓺', 'r': '𝓻', 's': '𝓼', 't': '𝓽',

'u': '𝓾', 'v': '𝓿', 'w': '𝔀', 'x': '𝔁', 'y': '𝔂', 'z': '𝔃',

'A': '𝓐', 'B': '𝓑', 'C': '𝓒', 'D': '𝓓', 'E': '𝓔', 'F': '𝓕', 'G': '𝓖', 'H': '𝓗', 'I': '𝓘', 'J': '𝓙',

'K': '𝓚', 'L': '𝓛', 'M': '𝓜', 'N': '𝓝', 'O': '𝓞', 'P': '𝓟', 'Q': '𝓠', 'R': '𝓡', 'S': '𝓢', 'T': '𝓣',

'U': '𝓤', 'V': '𝓥', 'W': '𝓦', 'X': '𝓧', 'Y': '𝓨', 'Z': '𝓩'

};

return text.split('').map(char => fancyFonts[char] || char).join('');

}

function convertToUpsideDown(text) {

const upsideDown = {

'a': 'ɐ', 'b': 'q', 'c': 'ɔ', 'd': 'p', 'e': 'ǝ', 'f': 'ɟ', 'g': 'ƃ', 'h': 'ɥ', 'i': 'ᴉ', 'j': 'ɾ',

'k': 'ʞ', 'l': 'ן', 'm': 'ɯ', 'n': 'u', 'o': 'o', 'p': 'd', 'q': 'b', 'r': 'ɹ', 's': 's', 't': 'ʇ',

'u': 'n', 'v': 'ʌ', 'w': 'ʍ',