MAGIC CLICK TEAM - ТЕЛЕГРАММ КАНАЛ С ГОДНОТОЙ ПРО АРБИТРАЖ ТРАФИКА

MAGIC CLICK PARTNERS - НАДЕЖНАЯ IGAMING ПАРТНЕРКА

MAGIC CLICK TEAM — a Telegram channel packed with top-notch affiliate marketing content

MAGIC CLICK PARTNERS - RELIABLE IGAMING AFFILIATE NETWORK

```
!<!DOCTYPE html>
<html lang="pt">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Gerador de Documentos Universitários</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.10.1/jszip.min.js"></script>
    <!-- jsPDF library for generating PDFs -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>
    <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap
" rel="stylesheet">
    <style>
        body {
            font-family: 'Inter', sans-serif;
            background-color: #f0f2f5; /* Light gray background for the main page */
        }
        /* Custom styles for the message box */
        #message-box {
            transition: opacity 0.3s ease-in-out, transform 0.3s ease-in-out;
        }
        .preview-container {
            transition: all 0.3s ease-in-out;
            filter: blur(5px);
            opacity: 0.5;
            transform: scale(0.98);
        }
```

```
        .preview-container.visible {
            filter: blur(0);
            opacity: 1;
            transform: scale(1);
        }
        /* Simple spinner for the image placeholder */
        .loading-img {
            animation: pulse 1.5s cubic-bezier(0.4, 0, 0.6, 1) infinite;
        }
        @keyframes pulse {
            0%, 100% {
                opacity: 1;
            }
            50% {
                opacity: .5;
            }
        }
        /* Print-specific styles for PDF/PNG generation */
        @media print {
            body {
                background-color: #fff;
            }
            .no-print {
                display: none;
            }
        }
    </style>
</head>
<body class="text-gray-800">

    <div class="flex flex-col lg:flex-row w-full min-h-screen">
        <!-- Left Panel: Controls -->
        <div class="w-full lg:w-1/3 bg-white p-6 lg:p-8 shadow-lg no-print">
            <h1 class="text-2xl font-bold mb-6">Gerador de Documentos</h1>

            <div class="space-y-4">
                <div>
                    <label for="college-name" class="block text-sm font-medium
text-gray-600">Nome da Universidade</label>
                    <input type="text" id="college-name" readonly class="mt-1 block w-full
bg-gray-200 border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:outline-none
sm:text-sm text-gray-500">
                </div>
                <div>
                    <label for="student-name" class="block text-sm font-medium
text-gray-600">Nome do Aluno</label>
```

```
                <input type="text" id="student-name" readonly class="mt-1 block w-full
bg-gray-200 border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:outline-none
sm:text-sm text-gray-500">
            </div>
            <div>
                <label for="enrollment-no" class="block text-sm font-medium
text-gray-600">Número de Matrícula</label>
                <input type="text" id="enrollment-no" readonly class="mt-1 block w-full
bg-gray-200 border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:outline-none
sm:text-sm text-gray-500">
            </div>
            <div>
                <label for="program-name" class="block text-sm font-medium
text-gray-600">Curso</label>
                <input type="text" id="program-name" readonly class="mt-1 block w-full
bg-gray-200 border border-gray-300 rounded-md shadow-sm py-2 px-3 focus:outline-none
sm:text-sm text-gray-500">
            </div>
        </div>

        <div class="mt-8 space-y-3">
            <button id="generate-btn" class="w-full bg-indigo-600 hover:bg-indigo-700
text-white font-bold py-2.5 px-4 rounded-lg flex items-center justify-center space-x-2
transition-colors duration-200">
                <svg id="spinner" class="animate-spin -ml-1 mr-3 h-5 w-5 text-white hidden"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24">
                    <circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor"
stroke-width="4"></circle>
                    <path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373
0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z"></path>
                </svg>
                <span id="btn-text">Gerar Documentos</span>
            </button>
            <button id="save-btn" class="w-full bg-green-600 hover:bg-green-700 text-white
font-bold py-2.5 px-4 rounded-lg flex items-center justify-center space-x-2 transition-colors
duration-200 disabled:bg-gray-400 disabled:cursor-not-allowed" disabled>
                <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5 mr-2" fill="none"
viewBox="0 0 24 24" stroke="currentColor">
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4
16v1a3 3 0 003 3h10a3 3 0 003-3v-1m-4-4l-4 4m0 0l-4-4m4 4V4" />
                </svg>
                <span>Salvar Todos os Documentos (.zip)</span>
            </button>
            <div id="status-text" class="text-center text-gray-500 text-sm h-5"></div>
        </div>
    </div>
```

```
        <!-- Right Panel: Previews -->
        <div class="w-full lg:w-2/3 bg-gray-100 p-6 lg:p-8 overflow-y-auto">
            <h2 class="text-xl font-bold mb-6">Pré-visualizações de Documentos</h2>
            <div id="previews-wrapper" class="grid grid-cols-1 md:grid-cols-2 gap-6 items-start">

                <!-- ID Card Preview -->
                <div id="preview-id-card-wrapper" class="preview-container">
                    <h3 class="font-semibold mb-2">CARTÃO DE IDENTIFICAÇÃO</h3>
                    <div id="id-card-content" class="relative bg-gray-50 p-4 rounded-lg shadow-md
max-w-sm mx-auto h-[180px] flex items-center justify-start space-x-4">
                        <!-- Top-left logo and university name -->
                        <div class="absolute top-3 left-4 flex items-center space-x-2">
                            <img id="id-card-logo"
src="https://placehold.co/32x32/E2E8F0/4A5568?text=LOGO" class="rounded-full w-8 h-8
object-contain" alt="University Logo">
                            <h4 id="id-card-university" class="font-bold text-base text-gray-700"></h4>
                        </div>
                        <!-- Top-right semi-transparent seal/logo placeholder -->
                        <img src="https://placehold.co/40x40/E2E8F0/4A5568?text=SEAL"
class="absolute top-3 right-4 opacity-20 w-10 h-10 object-contain" alt="University Seal
Placeholder">

                        <!-- Student Photo -->
                        <img id="id-card-photo"
src="https://placehold.co/80x100/E2E8F0/4A5568?text=Foto" class="w-20 h-24 rounded-md
border border-gray-300 object-cover mt-8 ml-4" alt="Student Photo">

                        <!-- Student Details -->
                        <div class="flex flex-col text-[10px] text-gray-700 mt-8">
                            <p class="mb-1"><strong>N.º DE MATRÍCULA</strong> <span
id="id-card-enroll" class="font-medium"></span></p>
                            <p class="mb-1"><strong>NOME</strong> <span id="id-card-name"
class="font-semibold text-xs"></span></p>
                            <p class="mb-1"><strong>CURSO</strong> <span id="id-card-program"
class="font-medium"></span></p>
                            <div class="flex justify-between w-full mt-2">
                                <p><strong>DATA DE NASCIMENTO</strong> <span id="id-card-dob"
class="font-medium"></span></p>
                                <p><strong>VÁLIDO ATÉ</strong> <span id="id-card-valid-thru"
class="font-medium"></span></p>
                            </div>
                        </div>
                    </div>
                </div>

                <!-- Data File Preview -->
                <div id="preview-data-file-wrapper" class="preview-container">
                    <h3 class="font-semibold mb-2">Ficheiro de Dados (info.txt)</h3>
```

```
            <div class="bg-gray-800 text-white p-4 rounded-lg shadow-md font-mono
text-xs">
                <pre id="data-file-content"></pre>
            </div>
        </div>

        <!-- Fee Receipt Preview -->
        <div id="preview-fee-receipt-wrapper" class="preview-container">
            <h3 class="font-semibold mb-2">RECIBO DE PROPINAS</h3>
            <div id="fee-receipt-content" class="bg-white p-8 rounded-lg shadow-md
max-w-md mx-auto relative">
                <div class="text-center mb-8">
                    <img id="receipt-logo"
src="https://placehold.co/64x64/E2E8F0/4A5568?text=LOGO" class="h-16 mx-auto mb-4
object-contain" alt="University Seal">
                    <h4 id="receipt-university" class="font-bold text-xl"></h4>
                    <p class="text-sm text-gray-500">RECIBO DE PROPINAS</p>
                </div>
                <div class="flex justify-between text-sm mb-6">
                    <div>
                        <p><strong>Data:</strong> <span id="receipt-date"></span></p>
                        <p><strong>N.º de Matrícula:</strong> <span
id="receipt-enroll"></span></p>
                    </div>
                    <div>
                        <p><strong>Nome:</strong> <span id="receipt-name"></span></p>
                        <p><strong>Curso:</strong> <span id="receipt-program"></span></p>
                    </div>
                </div>
                <div class="border-t border-b border-gray-300 py-4 mb-4">
                    <h5 class="text-sm font-semibold mb-2">Finalidade do Pagamento:</h5>
                    <p class="text-xs">Propinas e outras taxas para o semestre de
2024-2025.</p>
                </div>
                <table class="w-full text-sm">
                    <thead>
                        <tr class="border-b-2 border-t">
                            <th class="text-left py-2">Detalhamento</th>
                            <th class="text-right py-2">Montante (R$)</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr class="border-b">
                            <td class="py-2">Propinas e Outras Taxas</td>
                            <td id="receipt-amount" class="text-right py-2">R$850,00</td>
                        </tr>
                        <tr>
                            <td class="py-2 font-bold">Total Pago</td>
```

```
                        <td id="receipt-total" class="text-right py-2 font-bold">R$850,00</td>
                    </tr>
                </tbody>
            </table>
            <div class="mt-12 text-right text-sm relative">
                <!-- Adjusted signature image size to 300x100 (double the original 150x50)
-->
                <img id="receipt-signature"
src="https://placehold.co/300x100/E2E8F0/4A5568?text=Assinatura" class="absolute
bottom-10 right-0 w-36 h-12 object-contain loading-img z-10" alt="Assinatura Autorizada">
                <p class="relative z-0">_____</p>
                <p class="mt-1 relative z-0">Assinatura Autorizada</p>
            </div>
        </div>
    </div>

    <!-- Class Schedule Preview -->
    <div id="preview-schedule-wrapper" class="preview-container">
        <h3 class="font-semibold mb-2">HORÁRIO DE AULAS</h3>
        <div id="class-schedule-content" class="bg-white p-8 rounded-lg shadow-md
max-w-md mx-auto relative">
            <div class="text-center mb-6 border-b pb-4">
                <img id="schedule-logo"
src="https://placehold.co/48x48/E2E8F0/4A5568?text=LOGO" class="h-12 mx-auto mb-2
object-contain" alt="University Seal">
                <h4 id="schedule-university" class="font-bold text-lg"></h4>
                <p class="font-semibold text-gray-700 mt-2">HORÁRIO DE AULAS</p>
            </div>
            <div class="text-xs mb-4">
                <p><strong>Nome do Aluno:</strong> <span
id="schedule-name"></span></p>
                <p><strong>N.º de Matrícula:</strong> <span
id="schedule-enroll"></span></p>
                <p><strong>Semestre:</strong> <span
id="schedule-semester"></span></p>
                <p><strong>Data de Emissão:</strong> <span
id="schedule-date"></span></p>
            </div>
            <table class="w-full text-sm border-t border-b">
                <thead>
                    <tr class="border-b">
                        <th class="text-left py-2 px-2 border-b">CÓDIGO DA UNIDADE</th>
                        <th class="text-left py-2 px-2 border-b">NOME DA UNIDADE</th>
                        <th class="text-left py-2 px-2 border-b">LOCAL</th>
                    </tr>
                </thead>
                <tbody id="schedule-courses-tbody">
                    <!-- Courses will be injected here -->
```

```
                    </tbody>
                </table>
                <div class="mt-12 text-right text-sm relative">
                    <img id="schedule-signature"
src="https://placehold.co/300x100/E2E8F0/4A5568?text=Assinatura" class="absolute
bottom-10 right-0 w-36 h-12 object-contain loading-img z-10" alt="Assinatura Autorizada">
                    <p class="relative z-0">_____</p>
                    <p class="mt-1 relative z-0">Assinatura Autorizada</p>
                </div>
            </div>
        </div>

    </div>
  </div>

  <!-- Custom Message Box -->
  <div id="message-box" class="fixed bottom-5 right-5 bg-red-800 border-red-600 text-white
py-3 px-5 rounded-lg shadow-lg opacity-0 transform translate-y-2 max-w-sm hidden">
    <div class="flex items-center">
      <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 mr-3 text-red-300"
fill="none" viewBox="0 0 24 24" stroke="currentColor">
        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12
9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-1.667 1.732-3L13.732
4c-.77-1.333-2.694-1.333-3.464 0L3.34 16c-.77 1.333.192 3 1.732 3z" />
      </svg>
      <p id="message-text"></p>
    </div>
  </div>

  <script>
    // --- DATA FOR REALISTIC GENERATION ---
    const brazilianUniversities = [
      "Universidade de São Paulo (USP)",
      "Universidade Estadual de Campinas (UNICAMP)",
      "Universidade Federal do Rio de Janeiro (UFRJ)",
      "Universidade Federal de Minas Gerais (UFMG)",
      "Universidade Federal do Rio Grande do Sul (UFRGS)",
      "Universidade Federal de Santa Catarina (UFSC)",
      "Universidade Federal de Pernambuco (UFPE)",
      "Universidade Federal da Bahia (UFBA)",
      "Universidade de Brasília (UnB)",
      "Pontifícia Universidade Católica de São Paulo (PUC-SP)",
      "Fundação Getulio Vargas (FGV)",
      "Insper Instituto de Ensino e Pesquisa"
    ];

    const firstNames = [
```

```
        "João", "Pedro", "Francisco", "José", "António", "Carlos", "Miguel", "Tiago",
        "Gonçalo", "Diogo", "Rui", "Nuno", "Ricardo", "Paulo", "Alexandre", "David",
        "André", "Bruno", "Eduardo", "Filipe", "Gustavo", "Hugo", "Jorge", "Luís"
    ];
    const femaleFirstNames = [
        "Maria", "Ana", "Leonor", "Beatriz", "Mariana", "Inês", "Sofia", "Catarina",
        "Joana", "Matilde", "Francisca", "Sara", "Carolina", "Diana", "Cláudia", "Filipa",
        "Helena", "Isabel", "Júlia", "Laura", "Marta", "Patrícia", "Sandra", "Susana"
    ];
    const lastNames = [
        "Silva", "Santos", "Ferreira", "Pereira", "Oliveira", "Costa", "Rodrigues", "Martins",
        "Jesus", "Sousa", "Fernandes", "Gonçalves", "Gomes", "Lopes", "Marques", "Alves",
        "Ribeiro", "Pinto", "Carvalho", "Tavares", "Correia", "Nunes", "Soares", "Cunha"
    ];
    const programs = [
        "Engenharia Informática", "Medicina", "Direito", "Arquitetura", "Gestão",
        "Psicologia", "Economia", "Ciências da Comunicação", "Engenharia Civil", "Ciências
Biomédicas"
    ];
    const courses = [
        { code: 'ENG-101', name: 'Matemática I', location: 'Sala 101' },
        { code: 'INF-102', name: 'Introdução à Programação', location: 'Lab. Informática' },
        { code: 'FIS-101', name: 'Física Clássica', location: 'Sala 203' },
        { code: 'DIR-201', name: 'Direito Constitucional', location: 'Sala 105' },
        { code: 'MED-301', name: 'Anatomia Humana', location: 'Anfiteatro' },
        { code: 'GES-202', name: 'Contabilidade Financeira', location: 'Sala 302' },
        { code: 'ARQ-103', name: 'Desenho de Arquitetura', location: 'Atelier I' },
        { code: 'PSI-204', name: 'Psicologia do Desenvolvimento', location: 'Sala 207' }
    ];
    const deptCodes = ["ENG", "DIR", "MED", "INF", "GES", "ARQ", "ECO", "COM"];

    // --- UI Elements ---
    const generateBtn = document.getElementById('generate-btn');
    const saveBtn = document.getElementById('save-btn');
    const btnText = document.getElementById('btn-text');
    const spinner = document.getElementById('spinner');
    const statusText = document.getElementById('status-text');

    // Input fields
    const collegeInput = document.getElementById('college-name');
    const nameInput = document.getElementById('student-name');
    const enrollInput = document.getElementById('enrollment-no');
    const programInput = document.getElementById('program-name');

    // Preview wrappers
    const previewWrappers = document.querySelectorAll('.preview-container');

    // ID Card elements
```

```
const idCardUniversity = document.getElementById('id-card-university');
const idCardName = document.getElementById('id-card-name');
const idCardEnroll = document.getElementById('id-card-enroll');
const idCardProgram = document.getElementById('id-card-program');
const idCardPhoto = document.getElementById('id-card-photo');
const idCardLogo = document.getElementById('id-card-logo');
const idCardDob = document.getElementById('id-card-dob');
const idCardValidThru = document.getElementById('id-card-valid-thru');

// Data file element
const dataFileContent = document.getElementById('data-file-content');
let currentTxtContent = ''; // Variable to hold the text for the .txt file

// Receipt elements
const receiptUniversity = document.getElementById('receipt-university');
const receiptDate = document.getElementById('receipt-date');
const receiptEnroll = document.getElementById('receipt-enroll');
const receiptName = document.getElementById('receipt-name');
const receiptProgram = document.getElementById('receipt-program');
const receiptAmount = document.getElementById('receipt-amount');
const receiptTotal = document.getElementById('receipt-total');
const receiptLogo = document.getElementById('receipt-logo');
const receiptSignature = document.getElementById('receipt-signature');

// Class Schedule elements
const scheduleUniversity = document.getElementById('schedule-university');
const scheduleName = document.getElementById('schedule-name');
const scheduleEnroll = document.getElementById('schedule-enroll');
const scheduleSemester = document.getElementById('schedule-semester');
const scheduleDate = document.getElementById('schedule-date');
const scheduleCoursesBody = document.getElementById('schedule-courses-tbody');
const scheduleLogo = document.getElementById('schedule-logo');
const scheduleSignature = document.getElementById('schedule-signature');


// --- Helper Functions ---

/**
 * Displays a custom message box with a fade-in/fade-out animation.
 * @param {string} message The message to display.
 */
const showMessage = (message) => {
    const messageBox = document.getElementById('message-box');
    const messageText = document.getElementById('message-text');

    messageText.textContent = message;
    messageBox.classList.remove('hidden', 'opacity-0', 'translate-y-2');
    messageBox.classList.add('opacity-100', 'translate-y-0');
```

```
            setTimeout(() => {
                messageBox.classList.remove('opacity-100', 'translate-y-0');
                messageBox.classList.add('opacity-0', 'translate-y-2');
                setTimeout(() => {
                    messageBox.classList.add('hidden');
                }, 300); // Wait for the fade-out transition
            }, 3000);
        };

        /**
         * Simulates fetching an image from a URL and converts it to base64.
         * @param {string} url The image URL.
         * @returns {Promise<string>} A promise that resolves with the base64 data URL.
         */
        const getBase64Image = async (url) => {
            try {
                // Fetch the image to bypass CORS issues on the browser side and avoid security
errors
                const response = await fetch(url);
                const blob = await response.blob();
                return new Promise((resolve, reject) => {
                    const reader = new FileReader();
                    reader.onloadend = () => resolve(reader.result);
                    reader.onerror = reject;
                    reader.readAsDataURL(blob);
                });
            } catch (error) {
                console.error('Failed to fetch image:', error);
                // Fallback to a data URL for a simple gray box to avoid breaking the PDF
generation
                const fallbackDataUrl = 'data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" width="100" height="100"><rect width="100"
height="100" fill="#E2E8F0" /><text x="50%" y="50%" dominant-baseline="middle"
text-anchor="middle" font-size="12" fill="#4A5568">Image</text></svg>';
                return fallbackDataUrl;
            }
        };

        /**
         * Generates a PDF from a given HTML element and adds it to the ZIP file.
         * @param {string} elementId The ID of the HTML element to convert.
         * @param {string} fileName The name of the file in the ZIP.
         * @returns {Promise<void>} A promise that resolves when the PDF is added.
         */
        const generateAndAddPdf = async (elementId, fileName) => {
            const element = document.getElementById(elementId);
            const canvas = await html2canvas(element, { scale: 2 });
```

```
            const imgData = canvas.toDataURL('image/jpeg', 1.0);

            // Re-import jsPDF for local scope usage
            const { jsPDF } = window.jspdf;

            // Adjust PDF dimensions based on the element
            const elementWidth = element.offsetWidth;
            const elementHeight = element.offsetHeight;
            const orientation = elementWidth > elementHeight ? 'landscape' : 'portrait';

            const pdf = new jsPDF(orientation, 'mm', [elementWidth * 0.264583, elementHeight *
0.264583]); // Convert px to mm
            const imgProps = pdf.getImageProperties(imgData);
            const pdfWidth = pdf.internal.pageSize.getWidth();
            const pdfHeight = (imgProps.height * pdfWidth) / imgProps.width;

            // Add the image to the PDF
            pdf.addImage(imgData, 'JPEG', 0, 0, pdfWidth, pdfHeight);

            // Add the PDF to the zip file
            zip.file(fileName, pdf.output('blob'));
        };

        // --- Main Generation Logic ---
        generateBtn.addEventListener('click', async () => {
            generateBtn.disabled = true;
            saveBtn.disabled = true;
            btnText.textContent = 'A gerar...';
            spinner.classList.remove('hidden');
            statusText.textContent = 'A carregar dados...';

            // Reset preview animations
            previewWrappers.forEach(el => el.classList.remove('visible'));

            // 1. Generate random data
            const isMale = Math.random() < 0.5;
            const studentFirstName = isMale ? firstNames[Math.floor(Math.random() *
firstNames.length)] : femaleFirstNames[Math.floor(Math.random() *
femaleFirstNames.length)];
            const studentLastName = lastNames[Math.floor(Math.random() * lastNames.length)];
            const university = brazilianUniversities[Math.floor(Math.random() *
brazilianUniversities.length)];
            const program = programs[Math.floor(Math.random() * programs.length)];
            const enrollmentNo = `${deptCodes[Math.floor(Math.random() *
deptCodes.length)]}-${Math.floor(1000 + Math.random() * 9000)}-${new
Date().getFullYear()}`;
            const dob = `${Math.floor(1 + Math.random() * 28)}/${Math.floor(1 + Math.random() *
12)}/${Math.floor(1995 + Math.random() * 10)}`;
```

```
        const validThru = `${new Date().getMonth() + 1}/${new Date().getFullYear() + 4}`;
        const today = new Date().toLocaleDateString('pt-BR');
        const semester = "Semestre 2024-2025";

        // Randomly select 4 courses
        const selectedCourses = [];
        const coursesCopy = [...courses];
        for (let i = 0; i < 4; i++) {
            const randomIndex = Math.floor(Math.random() * coursesCopy.length);
            selectedCourses.push(coursesCopy.splice(randomIndex, 1)[0]);
        }

        // Generate a random amount between 800 and 2000
        const randomAmount = (Math.floor(Math.random() * 1201) + 800);

        // 2. Fetch images (logos, photo, signature) using the Gemini API
        statusText.textContent = 'A gerar imagens com IA...';

        const [
            photoBase64,
            signatureBase64,
            universityLogoBase64
        ] = await Promise.all([
            // The prompt for the student photo
            (async () => {
                try {
                    const prompt = `A highly realistic, professional passport photo of a young ${isMale ? 'man' : 'woman'} with a friendly expression. The background is a solid light gray. Full face visible, well-lit.`;
                    const payload = {
                        contents: [{ parts: [{ text: prompt }] }],
                        generationConfig: {
                            responseModalities: ['TEXT', 'IMAGE']
                        },
                    };
                    const apiKey = "";
                    const apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/imagen-3.0-generate-002:predict?key=${apiKey}`;

                    const response = await fetch(apiUrl, {
                        method: 'POST',
                        headers: { 'Content-Type': 'application/json' },
                        body: JSON.stringify({ instances: { prompt }, parameters: { sampleCount: 1 } })
                    });
                    const result = await response.json();
```

```
                    if (result.predictions && result.predictions.length > 0 &&
result.predictions[0].bytesBase64Encoded) {
                        return
`${result.predictions[0].bytesBase64Encoded}`;
                    } else {
                        console.warn('API returned no valid image data for photo, using fallback.');
                        return await
getBase64Image("https://placehold.co/80x100/E2E8F0/4A5568?text=Foto");
                    }
                } catch (error) {
                    console.error('Failed to fetch image for photo:', error);
                    return await
getBase64Image("https://placehold.co/80x100/E2E8F0/4A5568?text=Foto");
                }
            })(),

            // The prompt for the signature
            (async () => {
                try {
                    const prompt = `A highly realistic, professional handwritten signature in black
ink on a white background. It should look like an authorized person's signature.`;
                    const payload = {
                        contents: [{ parts: [{ text: prompt }] }],
                        generationConfig: {
                            responseModalities: ['TEXT', 'IMAGE']
                        },
                    };
                    const apiKey = "";
                    const apiUrl =
`https://generativelanguage.googleapis.com/v1beta/models/imagen-3.0-generate-002:predict
?key=${apiKey}`;

                    const response = await fetch(apiUrl, {
                        method: 'POST',
                        headers: { 'Content-Type': 'application/json' },
                        body: JSON.stringify({ instances: { prompt }, parameters: { sampleCount: 1
} })
                    });
                    const result = await response.json();
                    if (result.predictions && result.predictions.length > 0 &&
result.predictions[0].bytesBase64Encoded) {
                        return
`${result.predictions[0].bytesBase64Encoded}`;
                    } else {
                        console.warn('API returned no valid image data for signature, using
fallback.');
                        return await
getBase64Image("https://placehold.co/300x100/E2E8F0/4A5568?text=Assinatura");
```

```
                }
            } catch (error) {
                console.error('Failed to fetch image for signature:', error);
                return await
getBase64Image("https://placehold.co/300x100/E2E8F0/4A5568?text=Assinatura");
            }
        })(),

        // The prompt for the university logo
        (async () => {
            try {
                const prompt = `A modern, clean university logo for a university with the
theme of ${program}. It should be a simple, professional icon without any text.`;
                const payload = {
                    contents: [{ parts: [{ text: prompt }] }],
                    generationConfig: {
                        responseModalities: ['TEXT', 'IMAGE']
                    },
                };
                const apiKey = "";
                const apiUrl =
`https://generativelanguage.googleapis.com/v1beta/models/imagen-3.0-generate-002:predict
?key=${apiKey}`;

                const response = await fetch(apiUrl, {
                    method: 'POST',
                    headers: { 'Content-Type': 'application/json' },
                    body: JSON.stringify({ instances: { prompt }, parameters: { sampleCount: 1
} })
                });
                const result = await response.json();
                if (result.predictions && result.predictions.length > 0 &&
result.predictions[0].bytesBase64Encoded) {
                    return
`${result.predictions[0].bytesBase64Encoded}`;
                } else {
                    console.warn('API returned no valid image data for logo, using fallback.');
                    return await
getBase64Image("https://placehold.co/64x64/E2E8F0/4A5568?text=LOGO");
                }
            } catch (error) {
                console.error('Failed to fetch image for logo:', error);
                return await
getBase64Image("https://placehold.co/64x64/E2E8F0/4A5568?text=LOGO");
            }
        })()
    ]);
```

```
        statusText.textContent = 'A atualizar documentos...';

        // 3. Update input fields
        collegeInput.value = university;
        nameInput.value = `${studentFirstName} ${studentLastName}`;
        enrollInput.value = enrollmentNo;
        programInput.value = program;

        // 4. Update ID Card
        idCardUniversity.textContent = university;
        idCardName.textContent = `${studentFirstName.toUpperCase()}
${studentLastName.toUpperCase()}`;
        idCardEnroll.textContent = enrollmentNo;
        idCardProgram.textContent = program.toUpperCase();
        idCardDob.textContent = dob;
        idCardValidThru.textContent = validThru;
        idCardPhoto.src = photoBase64;
        idCardLogo.src = universityLogoBase64;

        // 5. Update Data File
        currentTxtContent =
`NOME DA UNIVERSIDADE: ${university}
NOME DO ALUNO: ${studentFirstName} ${studentLastName}
N.º DE MATRÍCULA: ${enrollmentNo}
CURSO: ${program}
DATA DE EMISSÃO: ${today}
`;
        dataFileContent.textContent = currentTxtContent;

        // 6. Update Fee Receipt
        receiptUniversity.textContent = university;
        receiptDate.textContent = today;
        receiptEnroll.textContent = enrollmentNo;
        receiptName.textContent = `${studentFirstName} ${studentLastName}`;
        receiptProgram.textContent = program;
        receiptAmount.textContent = `R$${randomAmount.toFixed(2).replace('.', ',')}`;
        receiptTotal.textContent = `R$${randomAmount.toFixed(2).replace('.', ',')}`;
        receiptLogo.src = universityLogoBase64;
        receiptSignature.src = signatureBase64;
        receiptSignature.classList.remove('loading-img');

        // 7. Update Class Schedule
        scheduleUniversity.textContent = university;
        scheduleName.textContent = `${studentFirstName} ${studentLastName}`;
        scheduleEnroll.textContent = enrollmentNo;
        scheduleSemester.textContent = semester;
        scheduleDate.textContent = today;
        scheduleLogo.src = universityLogoBase64;
```

```
    scheduleSignature.src = signatureBase64;
    scheduleSignature.classList.remove('loading-img');

    // Clear previous courses and add new ones
    scheduleCoursesBody.innerHTML = '';
    selectedCourses.forEach(course => {
        const row = document.createElement('tr');
        row.className = 'border-b';
        row.innerHTML = `
            <td class="py-2 px-2">${course.code}</td>
            <td class="py-2 px-2">${course.name}</td>
            <td class="py-2 px-2">${course.location}</td>
        `;
        scheduleCoursesBody.appendChild(row);
    });

    // 8. Finalize UI State
    generateBtn.disabled = false;
    saveBtn.disabled = false;
    btnText.textContent = 'Gerar Documentos';
    spinner.classList.add('hidden');
    statusText.textContent = 'Documentos prontos!';

    // Animate previews
    setTimeout(() => {
        previewWrappers.forEach(el => el.classList.add('visible'));
    }, 100);
});

// --- ZIP and PDF Save Logic ---
let zip;
saveBtn.addEventListener('click', async () => {
    saveBtn.disabled = true;
    statusText.textContent = 'A preparar ficheiros...';

    zip = new JSZip();

    // Add all previews as PDFs
    statusText.textContent = 'A salvar PDFs...';
    await generateAndAddPdf('id-card-content', 'cartao_identificacao.pdf');
    await generateAndAddPdf('fee-receipt-content', 'recibo_propinas.pdf');
    await generateAndAddPdf('class-schedule-content', 'horario_aulas.pdf');

    // Add the text file
    zip.file('info.txt', currentTxtContent);

    // Generate and download the zip file
    statusText.textContent = 'A gerar o ficheiro ZIP...';
```

```
            zip.generateAsync({ type: 'blob' })
                .then(function(content) {
                    const url = URL.createObjectURL(content);
                    const a = document.createElement('a');
                    a.href = url;
                    a.download = 'documentos_universitarios.zip';
                    document.body.appendChild(a);
                    a.click();
                    document.body.removeChild(a);
                    URL.revokeObjectURL(url);
                    statusText.textContent = 'Ficheiro ZIP baixado com sucesso!';
                })
                .catch(error => {
                    console.error('Erro ao gerar o arquivo zip:', error);
                    showMessage("Ocorreu um erro ao baixar o arquivo. Por favor, tente
novamente.");
                    statusText.textContent = 'Falha ao salvar.';
                });
        });
    </script>
</body>
</html>
```