<!DOCTYPE html>

<html lang="pt-BR">

<head>

<meta charset="UTF-8">

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

<title>UBS App - Gestão em Tempo Real</title>

<script src="https://cdn.tailwindcss.com"></script>

<script type="module" src="https://www.gstatic.com/firebasejs/10.12.3/firebase-app.js"></script>

<script type="module" src="https://www.gstatic.com/firebasejs/10.12.3/firebase-auth.js"></script>

<script type="module" src="https://www.gstatic.com/firebasejs/10.12.3/firebase-firestore.js"></script>

<style>

body {

font-family: 'Inter', sans-serif;

background-color: #f3f4f6;

}

.bg-risco-verde { background-color: #d1fae5; border-left: 4px solid #10b981; }

.bg-risco-amarelo { background-color: #fffac1; border-left: 4px solid #f59e0b; }

.bg-risco-vermelho { background-color: #fee2e2; border-left: 4px solid #ef4444; }

.bg-risco-neutro { background-color: #e5e7eb; border-left: 4px solid #9ca3af; }

</style>

</head>

<body class="flex items-center justify-center min-h-screen p-4 md:p-8">

<!-- Container Principal do Aplicativo com Layout de PC -->

<div class="flex flex-col md:flex-row bg-white rounded-xl shadow-lg w-full max-w-7xl h-[90vh]">

<!-- Sidebar de Navegação -->

<div id="sidebar" class="hidden md:flex flex-col p-6 space-y-4 bg-gray-100 rounded-l-xl shadow-inner w-full md:w-64">

<h1 class="text-xl font-bold text-gray-800 mb-4">Menu</h1>

<button id="acompanhamento-btn" class="hidden w-full bg-blue-600 text-white font-semibold p-3 rounded-lg hover:bg-blue-700 transition duration-200 text-left">

Ver Acompanhamento

</button>

<button id="historico-btn" class="hidden w-full bg-gray-500 text-white font-semibold p-3 rounded-lg hover:bg-gray-600 transition duration-200 text-left">

Ver Histórico

</button>

<button id="relatorios-btn" class="hidden w-full bg-green-600 text-white font-semibold p-3 rounded-lg hover:bg-green-700 transition duration-200 text-left">

Ver Relatórios

</button>

<button id="gerenciar-btn" class="hidden w-full bg-purple-600 text-white font-semibold p-3 rounded-lg hover:bg-purple-700 transition duration-200 text-left">

Gerenciar Usuários

</button>

<button id="logout-btn" class="hidden w-full bg-gray-400 text-white font-semibold p-3 rounded-lg hover:bg-gray-500 transition duration-200 text-left mt-auto">

Sair

</button>

</div>

<!-- Conteúdo Principal -->

<div id="main-content" class="flex-grow p-8 rounded-xl w-full">

<!-- Tela de Login -->

<div id="login-page" class="transition-opacity duration-500 ease-in-out h-full flex flex-col justify-center items-center">

<div class="w-full max-w-sm">

<h1 class="text-3xl font-bold text-gray-800 mb-2 text-center">Unidade Básica de Saúde</h1>

<h2 class="text-xl font-semibold text-gray-700 mb-6 text-center">ALMIR DE ALMEIDA - BOA VISTA</h2>

<p class="text-gray-600 text-center mb-8">Acesse com seu usuário e senha.</p>

<form id="login-form" class="space-y-4">

<div>

<label for="username" class="block text-sm font-medium text-gray-700">Usuário</label>

<input type="text" id="username" name="username" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200">

</div>

<div>

<label for="password" class="block text-sm font-medium text-gray-700">Senha</label>

<input type="password" id="password" name="password" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200">

</div>

<button type="submit" class="w-full bg-blue-600 text-white font-semibold p-3 rounded-lg hover:bg-blue-700 transition duration-200 mt-6">

Entrar

</button>

<p id="error-message" class="text-red-500 text-center hidden">Usuário ou senha incorretos.</p>

</form>

</div>

</div>

<!-- Tela de Acompanhamento -->

<div id="acompanhamento-page" class="hidden transition-opacity duration-500 ease-in-out h-full">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Acompanhamento de Pacientes</h1>

<p class="text-gray-600 text-center mb-8">Status de todos os pacientes na unidade.</p>

<div class="flex justify-between items-center mb-4">

<h2 class="text-xl font-bold text-gray-700">Pacientes em Fila</h2>

<button id="add-new-patient-btn" class="hidden bg-blue-600 text-white font-semibold py-2 px-4 rounded-lg hover:bg-blue-700 transition duration-200">

+ Novo Acolhimento

</button>

</div>

<div id="acompanhamento-list-container" class="bg-gray-100 p-4 rounded-xl shadow-inner max-h-[60vh] overflow-y-auto">

<ul id="pacientes-list" class="space-y-4">

<p id="no-patients-message" class="text-center text-gray-500">Nenhum paciente na fila.</p>

</ul>

</div>

</div>

<!-- Tela de Acolhimento (Recepção) -->

<div id="acolhimento-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Acolhimento</h1>

<p class="text-gray-600 text-center mb-8">Preencha os dados do paciente para iniciar a triagem.</p>

<form id="patient-form" class="grid grid-cols-1 md:grid-cols-2 gap-4">

<div><label for="nome" class="block text-sm font-medium text-gray-700">Nome Completo</label><input type="text" id="nome" name="nome" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="cartaoSus" class="block text-sm font-medium text-gray-700">Cartão do SUS</label><input type="text" id="cartaoSus" name="cartaoSus" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="cpf" class="block text-sm font-medium text-gray-700">CPF</label><input type="text" id="cpf" name="cpf" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="contato" class="block text-sm font-medium text-gray-700">Contato do Paciente</label><input type="tel" id="contato" name="contato" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200" placeholder="(XX) XXXXX-XXXX"></div>

<div><label for="dataNascimento" class="block text-sm font-medium text-gray-700">Data de Nascimento</label><input type="date" id="dataNascimento" name="dataNascimento" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="agenteSaude" class="block text-sm font-medium text-gray-700">Nome da Agente de Saúde</label><input type="text" id="agenteSaude" name="agenteSaude" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div>

<label for="equipe" class="block text-sm font-medium text-gray-700">Equipe</label>

<select id="equipe" name="equipe" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200">

<option value="">Selecione a Equipe</option>

<option value="Equipe 1 - Azul">Equipe 1 - Azul</option>

<option value="Equipe 2 - Verde">Equipe 2 - Verde</option>

<option value="Equipe 3 - Amarela">Equipe 3 - Amarela</option>

</select>

</div>

<div class="md:col-span-2">

<button type="submit" class="w-full bg-blue-600 text-white font-semibold p-3 rounded-lg hover:bg-blue-700 transition duration-200 mt-6">

Enviar para Pré-consulta

</button>

</div>

</form>

</div>

<!-- Tela de Pré-consulta (Técnico) -->

<div id="pre-consulta-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Pré-consulta</h1>

<p class="text-gray-600 text-center mb-4">Preencha os dados do paciente: <span id="pre-consulta-nome-paciente" class="font-bold text-gray-800"></span></p>

<form id="pre-consulta-form" class="grid grid-cols-1 md:grid-cols-2 gap-4">

<div><label for="hipertensao" class="block text-sm font-medium text-gray-700">Hipertenso</label><select id="hipertensao" name="hipertensao" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"><option value="">Selecione</option><option value="Sim">Sim</option><option value="Não">Não</option></select></div>

<div><label for="diabetes" class="block text-sm font-medium text-gray-700">Diabético</label><select id="diabetes" name="diabetes" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"><option value="">Selecione</option><option value="Sim">Sim</option><option value="Não">Não</option></select></div>

<div><label for="alergias" class="block text-sm font-medium text-gray-700">Alergias</label><input type="text" id="alergias" name="alergias" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200" placeholder="Ex: amendoim, dipirona, etc."></div>

<div><label for="situacaoEspecial" class="block text-sm font-medium text-gray-700">Situação Especial</label><select id="situacaoEspecial" name="situacaoEspecial" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"><option value="Nenhuma">Nenhuma</option><option value="Gestante">Gestante</option><option value="Idoso">Idoso</option><option value="Criança">Criança</option><option value="PCD">PCD</option></select></div>

<div><label for="classificacaoRisco" class="block text-sm font-medium text-gray-700">Classificação de Risco</label><select id="classificacaoRisco" name="classificacaoRisco" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"><option value="">Selecione</option><option value="Verde - Sem Urgência">Verde - Sem Urgência</option><option value="Amarelo - Atenção Imediata">Amarelo - Atenção Imediata</option><option value="Vermelho - Urgência Imediata">Vermelho - Urgência Imediata</option></select></div>

<div><label for="frequencia" class="block text-sm font-medium text-gray-700">Frequência Cardíaca (BPM)</label><input type="number" id="frequencia" name="frequencia" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="pressao" class="block text-sm font-medium text-gray-700">Pressão Arterial (Ex: 120/80 mmHg)</label><input type="text" id="pressao" name="pressao" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="peso" class="block text-sm font-medium text-gray-700">Peso (kg)</label><input type="number" step="0.1" id="peso" name="peso" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="altura" class="block text-sm font-medium text-gray-700">Altura (m)</label><input type="number" step="0.01" id="altura" name="altura" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div><label for="temperatura" class="block text-sm font-medium text-gray-700">Temperatura Corporal (ºC)</label><input type="number" step="0.1" id="temperatura" name="temperatura" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200"></div>

<div class="md:col-span-2">

<label for="queixas" class="block text-sm font-medium text-gray-700">Queixas e Informações</label>

<div class="flex space-x-2 mt-1">

<textarea id="queixas" name="queixas" rows="4" required class="flex-grow p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200" placeholder="Descreva os sintomas e queixas do paciente..."></textarea>

<button type="button" id="analyze-symptoms-btn" class="bg-purple-600 text-white font-semibold p-3 rounded-lg hover:bg-purple-700 transition duration-200 self-start">

✨ Analisar Sintomas

</button>

</div>

</div>

<div id="ai-analysis-container" class="md:col-span-2 hidden">

<label class="block text-sm font-medium text-gray-700">Análise do Gemini</label>

<div id="ai-analysis-box" class="mt-1 p-4 bg-gray-200 rounded-lg shadow-inner">

<p id="ai-analysis-text" class="text-sm text-gray-800 whitespace-pre-wrap">Aguardando análise...</p>

</div>

</div>

<div class="md:col-span-2">

<button type="submit" class="w-full bg-blue-600 text-white font-semibold p-3 rounded-lg hover:bg-blue-700 transition duration-200 mt-6">

Finalizar Pré-consulta

</button>

</div>

</form>

</div>

<!-- Tela de Acesso do Profissional (Médico/Enfermeiro) -->

<div id="professional-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Visualização do Profissional</h1>

<p class="text-gray-600 text-center mb-4">Revise os dados e finalize o encaminhamento para o paciente: <span id="prof-nome-paciente" class="font-bold text-gray-800"></span></p>

<div class="p-6 bg-gray-100 rounded-xl mb-6">

<h2 class="text-xl font-bold text-gray-800 mb-4">Resumo da Triagem</h2>

<div class="grid grid-cols-1 md:grid-cols-2 gap-y-2 gap-x-4 text-gray-700">

<h3 class="font-bold text-lg text-blue-800 col-span-2">Dados de Acolhimento</h3>

<p><strong>Nome Completo:</strong> <span id="prof-nome"></span></p>

<p><strong>Cartão do SUS:</strong> <span id="prof-cartaoSus"></span></p>

<p><strong>Data de Nascimento:</strong> <span id="prof-dataNascimento"></span></p>

<p><strong>Contato:</strong> <span id="prof-contato"></span></p>

<hr class="my-4 border-gray-300 col-span-2">

<h3 class="font-bold text-lg text-blue-800 col-span-2">Dados da Pré-consulta</h3>

<p><strong>Classificação de Risco:</strong> <span id="prof-classificacaoRisco"></span></p>

<p><strong>Queixas:</strong> <span id="prof-queixas"></span></p>

<p><strong>Análise do Gemini:</strong> <span id="prof-gemini-analysis" class="block mt-2 font-normal whitespace-pre-wrap"></span></p>

</div>

</div>

<form id="professional-form" class="space-y-4">

<div>

<label for="prof-observacoes" class="block text-sm font-medium text-gray-700">Observações do Profissional</label>

<textarea id="prof-observacoes" name="prof-observacoes" rows="4" class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200" placeholder="Insira aqui observações importantes..."></textarea>

</div>

<div class="flex flex-col md:flex-row space-y-4 md:space-y-0 md:space-x-4 mt-6">

<button type="submit" data-encaminhamento="Atendimento" class="w-full bg-green-600 text-white font-semibold p-3 rounded-lg hover:bg-green-700 transition duration-200">

Finalizar Atendimento

</button>

<button type="submit" data-encaminhamento="Acesso Garantido" class="w-full bg-orange-600 text-white font-semibold p-3 rounded-lg hover:bg-orange-700 transition duration-200">

Acesso Garantido

</button>

</div>

</form>

</div>

<!-- Telas de Visualização e Histórico -->

<div id="visualizar-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Detalhes do Atendimento</h1>

<p class="text-gray-600 text-center mb-4"><span id="visualizar-nome-paciente" class="font-bold text-gray-800"></span></p>

<div id="visualizar-content" class="p-6 bg-gray-100 rounded-xl mb-6 space-y-4 max-h-[60vh] overflow-y-auto"></div>

<button id="visualizar-back-btn" class="w-full bg-gray-400 text-white font-semibold p-3 rounded-lg hover:bg-gray-500 transition duration-200 mt-6">Voltar</button>

</div>

<div id="historico-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Histórico de Atendimentos</h1>

<p class="text-gray-600 text-center mb-8">Registros de todos os atendimentos finalizados.</p>

<div id="historico-list-container" class="bg-gray-100 p-4 rounded-xl shadow-inner max-h-[60vh] overflow-y-auto">

<ul id="historico-list" class="space-y-4">

<p id="no-history-message" class="text-center text-gray-500">Nenhum atendimento no histórico.</p>

</ul>

</div>

<button id="historico-back-btn" class="w-full bg-gray-400 text-white font-semibold p-3 rounded-lg hover:bg-gray-500 transition duration-200 mt-6">Voltar</button>

</div>

<!-- Tela de Relatórios -->

<div id="relatorios-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Relatórios</h1>

<p class="text-gray-600 text-center mb-8">Dados e estatísticas sobre os atendimentos.</p>

<div id="relatorios-container" class="space-y-6"></div>

<button id="relatorios-back-btn" class="w-full bg-gray-400 text-white font-semibold p-3 rounded-lg hover:bg-gray-500 transition duration-200 mt-6">Voltar</button>

</div>

<!-- Tela de Gerenciamento de Usuários (Gerente) -->

<div id="gerenciar-page" class="hidden transition-opacity duration-500 ease-in-out">

<h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">Gerenciar Usuários</h1>

<p class="text-gray-600 text-center mb-8">Crie e gerencie os usuários do sistema.</p>

<form id="create-user-form" class="space-y-4">

<div>

<label for="new-username" class="block text-sm font-medium text-gray-700">Nome de Usuário</label>

<input type="text" id="new-username" name="new-username" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200">

</div>

<div>

<label for="new-password" class="block text-sm font-medium text-gray-700">Senha</label>

<input type="password" id="new-password" name="new-password" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200">

</div>

<div>

<label for="new-role" class="block text-sm font-medium text-gray-700">Função</label>

<select id="new-role" name="new-role" required class="mt-1 block w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500 transition duration-200">

<option value="">Selecione a Função</option>

<option value="recepcao">Recepção</option>

<option value="tecnico">Técnico</option>

<option value="enfermeiro">Enfermeiro</option>

<option value="medico">Médico</option>

<option value="gerente">Gerente</option>

</select>

</div>

<button type="submit" class="w-full bg-blue-600 text-white font-semibold p-3 rounded-lg hover:bg-blue-700 transition duration-200 mt-6">

Criar Usuário

</button>

<p id="user-creation-message" class="text-sm text-center mt-2"></p>

</form>

<div id="user-list-container" class="mt-8 bg-gray-100 p-4 rounded-xl shadow-inner max-h-64 overflow-y-auto">

<h2 class="text-xl font-bold text-gray-800 mb-4 text-center">Usuários Cadastrados</h2>

<ul id="user-list" class="space-y-2"></ul>

</div>

<button id="gerenciar-back-btn" class="w-full bg-gray-400 text-white font-semibold p-3 rounded-lg hover:bg-gray-500 transition duration-200 mt-6">Voltar</button>

</div>

</div>

</div>

<script type="module">

import { initializeApp } from "https://www.gstatic.com/firebasejs/10.12.3/firebase-app.js";

import { getAuth, signInWithCustomToken, signInAnonymously } from "https://www.gstatic.com/firebasejs/10.12.3/firebase-auth.js";

import { getFirestore, collection, addDoc, getDocs, doc, onSnapshot, updateDoc, deleteDoc, setDoc } from "https://www.gstatic.com/firebasejs/10.12.3/firebase-firestore.js";

// Mapeamento de elementos do DOM

const loginPage = document.getElementById('login-page');

const acolhimentoPage = document.getElementById('acolhimento-page');

const preConsultaPage = document.getElementById('pre-consulta-page');

const professionalPage = document.getElementById('professional-page');

const acompanhamentoPage = document.getElementById('acompanhamento-page');

const historicoPage = document.getElementById('historico-page');

const relatoriosPage = document.getElementById('relatorios-page');

const visualizarPage = document.getElementById('visualizar-page');

const gerenciarPage = document.getElementById('gerenciar-page');

const visualizarContent = document.getElementById('visualizar-content');

const mainContent = document.getElementById('main-content');

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

const pacientesList = document.getElementById('pacientes-list');

const noPatientsMessage = document.getElementById('no-patients-message');

const historicoList = document.getElementById('historico-list');

const noHistoryMessage = document.getElementById('no-history-message');

const relatoriosContainer = document.getElementById('relatorios-container');

const userList = document.getElementById('user-list');

const userCreationMessage = document.getElementById('user-creation-message');

const acompanhamentoBtn = document.getElementById('acompanhamento-btn');

const historicoBtn = document.getElementById('historico-btn');

const relatoriosBtn = document.getElementById('relatorios-btn');

const gerenciarBtn = document.getElementById('gerenciar-btn');

const addNewPatientBtn = document.getElementById('add-new-patient-btn');

const logoutBtn = document.getElementById('logout-btn');

const historicoBackBtn = document.getElementById('historico-back-btn');

const relatoriosBackBtn = document.getElementById('relatorios-back-btn');

const visualizarBackBtn = document.getElementById('visualizar-back-btn');

const gerenciarBackBtn = document.getElementById('gerenciar-back-btn');

const loginForm = document.getElementById('login-form');

const patientForm = document.getElementById('patient-form');

const preConsultaForm = document.getElementById('pre-consulta-form');

const professionalForm = document.getElementById('professional-form');

const createUserForm = document.getElementById('create-user-form');

const errorMessage = document.getElementById('error-message');

const analyzeSymptomsBtn = document.getElementById('analyze-symptoms-btn');

const queixasTextarea = document.getElementById('queixas');

const aiAnalysisContainer = document.getElementById('ai-analysis-container');

const aiAnalysisText = document.getElementById('ai-analysis-text');

let currentUser = null;

let currentUserRole = null;

let pacienteSelecionadoId = null;

let aiAnalysis = '';

let appInitialized = false;

const appId = typeof \_\_app\_id !== 'undefined' ? \_\_app\_id : 'default-app-id';

const firebaseConfig = JSON.parse(typeof \_\_firebase\_config !== 'undefined' ? \_\_firebase\_config : '{}');

let db = null;

let auth = null;

let usersCollectionRef = null;

let pacientesCollectionRef = null;

let historicoCollectionRef = null;

// --- Firebase Initialization and Auth ---

async function initializeFirebase() {

if (appInitialized) return;

try {

const app = initializeApp(firebaseConfig);

db = getFirestore(app);

auth = getAuth(app);

const \_\_initial\_auth\_token = typeof \_\_initial\_auth\_token !== 'undefined' ? \_\_initial\_auth\_token : null;

if (\_\_initial\_auth\_token) {

await signInWithCustomToken(auth, \_\_initial\_auth\_token);

} else {

await signInAnonymously(auth);

}

const userId = auth.currentUser?.uid || crypto.randomUUID();

usersCollectionRef = collection(db, `artifacts/${appId}/public/data/users`);

pacientesCollectionRef = collection(db, `artifacts/${appId}/public/data/pacientes`);

historicoCollectionRef = collection(db, `artifacts/${appId}/public/data/historico`);

appInitialized = true;

setupRealtimeListeners();

} catch (error) {

console.error("Erro ao inicializar Firebase:", error);

// Opcionalmente, mostrar uma mensagem de erro na tela

}

}

// --- Realtime Listeners ---

function setupRealtimeListeners() {

if (!db) return;

// Escutando por mudanças na coleção de usuários

onSnapshot(usersCollectionRef, (snapshot) => {

let tempUsers = {};

snapshot.forEach(doc => {

tempUsers[doc.id] = doc.data();

});

renderUserList(tempUsers);

});

// Escutando por mudanças na coleção de pacientes

onSnapshot(pacientesCollectionRef, (snapshot) => {

let tempPacientes = [];

snapshot.forEach(doc => {

tempPacientes.push({ id: doc.id, ...doc.data() });

});

renderAcompanhamento(tempPacientes);

});

// Escutando por mudanças na coleção de histórico

onSnapshot(historicoCollectionRef, (snapshot) => {

let tempHistorico = [];

snapshot.forEach(doc => {

tempHistorico.push({ id: doc.id, ...doc.data() });

});

renderHistorico(tempHistorico);

});

}

// --- Funções Auxiliares de Renderização e Navegação ---

function showPage(pageId) {

const pages = [loginPage, acolhimentoPage, preConsultaPage, professionalPage, acompanhamentoPage, historicoPage, relatoriosPage, visualizarPage, gerenciarPage];

pages.forEach(page => page.classList.add('hidden'));

document.getElementById(pageId).classList.remove('hidden');

}

function resetApp() {

currentUser = null;

currentUserRole = null;

pacienteSelecionadoId = null;

document.getElementById('patient-form').reset();

document.getElementById('pre-consulta-form').reset();

document.getElementById('professional-form').reset();

sidebar.classList.add('hidden');

acompanhamentoBtn.classList.add('hidden');

historicoBtn.classList.add('hidden');

relatoriosBtn.classList.add('hidden');

gerenciarBtn.classList.add('hidden');

logoutBtn.classList.add('hidden');

addNewPatientBtn.classList.add('hidden');

showPage('login-page');

}

function getBgRiscoClass(risco) {

if (risco && risco.includes('Verde')) return 'bg-risco-verde border-l-4 border-green-500';

if (risco && risco.includes('Amarelo')) return 'bg-risco-amarelo border-l-4 border-yellow-500';

if (risco && risco.includes('Vermelho')) return 'bg-risco-vermelho border-l-4 border-red-500';

return 'bg-risco-neutro border-l-4 border-gray-400';

}

function renderAcompanhamento(pacientes) {

pacientesList.innerHTML = '';

if (pacientes.length === 0) {

noPatientsMessage.classList.remove('hidden');

} else {

noPatientsMessage.classList.add('hidden');

pacientes.forEach((paciente, index) => {

const li = document.createElement('li');

const bgClass = getBgRiscoClass(paciente.classificacaoRisco);

let statusText = paciente.status;

if (paciente.status === 'Atendimento Finalizado' && paciente.professional && paciente.professional.encaminhamento) {

statusText = `Finalizado: ${paciente.professional.encaminhamento}`;

}

let isClickable = false;

let clickAction = null;

if (currentUserRole === 'gerente') {

isClickable = true;

if (paciente.status === 'Aguardando Pré-consulta') {

clickAction = () => fillPreConsultaForm(paciente);

} else if (paciente.status === 'Aguardando Profissional') {

clickAction = () => fillProfessionalForm(paciente);

} else {

clickAction = () => fillVisualizarPage(paciente);

}

}

else if (currentUserRole === 'tecnico') {

if (paciente.status === 'Aguardando Pré-consulta') {

isClickable = true;

clickAction = () => fillPreConsultaForm(paciente);

} else if (paciente.status === 'Atendimento Finalizado') {

isClickable = true;

clickAction = () => fillVisualizarPage(paciente);

}

}

else if (currentUserRole === 'recepcao') {

if (paciente.status === 'Atendimento Finalizado') {

isClickable = true;

clickAction = () => fillVisualizarPage(paciente);

}

}

else if (['medico', 'enfermeiro'].includes(currentUserRole)) {

if (paciente.status === 'Aguardando Profissional') {

isClickable = true;

clickAction = () => fillProfessionalForm(paciente);

}

}

const cursorClass = isClickable ? 'cursor-pointer hover:opacity-80' : 'cursor-default';

li.innerHTML = `

<div class="p-4 rounded-lg shadow-sm flex items-center justify-between transition duration-200 ${bgClass} ${cursorClass}">

<div>

<h3 class="text-lg font-semibold text-gray-800">${paciente.nome}</h3>

<p class="text-sm text-gray-700">Chegada: ${paciente.horaChegada}</p>

</div>

<span class="text-sm font-medium text-gray-700">${statusText}</span>

</div>

`;

if (isClickable) {

li.querySelector('div').addEventListener('click', () => {

clickAction();

});

}

pacientesList.appendChild(li);

});

}

}

function renderHistorico(historico) {

historicoList.innerHTML = '';

if (historico.length === 0) {

noHistoryMessage.classList.remove('hidden');

} else {

noHistoryMessage.classList.add('hidden');

historico.forEach(atendimento => {

const li = document.createElement('li');

li.classList.add('p-4', 'bg-white', 'rounded-lg', 'shadow-sm', 'border-l-4', 'border-green-500', 'cursor-pointer');

li.innerHTML = `

<h3 class="text-lg font-bold text-gray-800">${atendimento.nome}</h3>

<p class="text-sm text-gray-500">Data: ${atendimento.acolhimento.timestamp}</p>

<p class="text-sm text-gray-700 mt-2">Profissional: ${atendimento.professional.nome}</p>

<p class="text-sm text-gray-700">Encaminhamento: ${atendimento.professional.encaminhamento}</p>

`;

li.addEventListener('click', () => {

fillVisualizarPage(atendimento);

});

historicoList.appendChild(li);

});

}

}

function renderUserList(users) {

userList.innerHTML = '';

Object.keys(users).forEach(username => {

const user = users[username];

const li = document.createElement('li');

li.classList.add('flex', 'justify-between', 'items-center', 'p-2', 'bg-white', 'rounded-lg', 'shadow-sm');

li.innerHTML = `

<span class="text-gray-800 font-semibold">${username}</span>

<span class="text-gray-600">(${user.role})</span>

`;

userList.appendChild(li);

});

}

function fillPreConsultaForm(paciente) {

pacienteSelecionadoId = paciente.id;

document.getElementById('pre-consulta-nome-paciente').textContent = paciente.nome;

preConsultaForm.reset();

aiAnalysisContainer.classList.add('hidden');

aiAnalysisText.textContent = 'Aguardando análise...';

aiAnalysis = '';

if (paciente.preConsulta) {

document.getElementById('hipertensao').value = paciente.preConsulta.hipertensao;

document.getElementById('diabetes').value = paciente.preConsulta.diabetes;

document.getElementById('alergias').value = paciente.preConsulta.alergias;

document.getElementById('situacaoEspecial').value = paciente.preConsulta.situacaoEspecial;

document.getElementById('classificacaoRisco').value = paciente.preConsulta.classificacaoRisco;

document.getElementById('frequencia').value = paciente.preConsulta.frequencia;

document.getElementById('pressao').value = paciente.preConsulta.pressao;

document.getElementById('peso').value = paciente.preConsulta.peso;

document.getElementById('altura').value = paciente.preConsulta.altura;

document.getElementById('temperatura').value = paciente.preConsulta.temperatura;

document.getElementById('queixas').value = paciente.preConsulta.queixas;

if (paciente.preConsulta.aiAnalysis) {

aiAnalysis = paciente.preConsulta.aiAnalysis;

aiAnalysisText.textContent = aiAnalysis;

aiAnalysisContainer.classList.remove('hidden');

}

}

showPage('pre-consulta-page');

}

function fillProfessionalForm(paciente) {

pacienteSelecionadoId = paciente.id;

document.getElementById('prof-nome-paciente').textContent = paciente.nome;

document.getElementById('prof-nome').textContent = paciente.acolhimento.nome;

document.getElementById('prof-cartaoSus').textContent = paciente.acolhimento.cartaoSus;

document.getElementById('prof-dataNascimento').textContent = paciente.acolhimento.dataNascimento;

document.getElementById('prof-contato').textContent = paciente.acolhimento.contato;

document.getElementById('prof-classificacaoRisco').textContent = paciente.preConsulta.classificacaoRisco;

document.getElementById('prof-queixas').textContent = paciente.preConsulta.queixas;

document.getElementById('prof-gemini-analysis').textContent = paciente.preConsulta.aiAnalysis || 'Nenhuma análise disponível.';

showPage('professional-page');

}

function fillVisualizarPage(paciente) {

document.getElementById('visualizar-nome-paciente').textContent = paciente.nome;

visualizarContent.innerHTML = `

<h3 class="font-bold text-lg text-blue-800">Dados de Acolhimento</h3>

<p><strong>Nome Completo:</strong> ${paciente.acolhimento.nome}</p>

<p><strong>Cartão do SUS:</strong> ${paciente.acolhimento.cartaoSus}</p>

<p><strong>Data de Nascimento:</strong> ${paciente.acolhimento.dataNascimento}</p>

<p><strong>Contato:</strong> ${paciente.acolhimento.contato}</p>

<p><strong>Agente de Saúde:</strong> ${paciente.acolhimento.agenteSaude}</p>

<p><strong>Equipe:</strong> ${paciente.acolhimento.equipe}</p>

<p><strong>Profissional de Acolhimento:</strong> ${paciente.acolhimento.profissionalNome}</p>

<p><strong>Horário:</strong> ${paciente.acolhimento.timestamp}</p>

<hr class="my-4 border-gray-300">

<h3 class="font-bold text-lg text-blue-800">Dados da Pré-consulta</h3>

${paciente.preConsulta ? `

<p><strong>Hipertenso:</strong> ${paciente.preConsulta.hipertensao}</p>

<p><strong>Diabético:</strong> ${paciente.preConsulta.diabetes}</p>

<p><strong>Alergias:</strong> ${paciente.preConsulta.alergias}</p>

<p><strong>Situação Especial:</strong> ${paciente.preConsulta.situacaoEspecial}</p>

<p><strong>Classificação de Risco:</strong> ${paciente.preConsulta.classificacaoRisco}</p>

<p><strong>Frequência Cardíaca:</strong> ${paciente.preConsulta.frequencia} BPM</p>

<p><strong>Pressão Arterial:</strong> ${paciente.preConsulta.pressao}</p>

<p><strong>Peso:</strong> ${paciente.preConsulta.peso} kg</p>

<p><strong>Altura:</strong> ${paciente.preConsulta.altura} m</p>

<p><strong>Temperatura Corporal:</strong> ${paciente.preConsulta.temperatura} ºC</p>

<p><strong>Queixas:</strong> ${paciente.preConsulta.queixas}</p>

<p><strong>Análise do Gemini:</strong> ${paciente.preConsulta.aiAnalysis || 'Nenhuma análise disponível.'}</p>

<p><strong>Profissional da Pré-consulta:</strong> ${paciente.preConsulta.profissionalNome}</p>

<p><strong>Horário:</strong> ${paciente.preConsulta.timestamp}</p>

` : '<p>Pré-consulta ainda não realizada.</p>'}

<hr class="my-4 border-gray-300">

<h3 class="font-bold text-lg text-blue-800">Conduta do Profissional</h3>

${paciente.professional ? `

<p><strong>Profissional:</strong> ${paciente.professional.nome}</p>

<p><strong>Registro:</strong> ${paciente.professional.registro}</p>

<p><strong>Encaminhamento:</strong> ${paciente.professional.encaminhamento}</p>

<p><strong>Observações:</strong> ${paciente.professional.observacoes}</p>

<p><strong>Horário:</strong> ${paciente.professional.timestamp}</p>

` : '<p>Atendimento profissional ainda não realizado.</p>'}

`;

showPage('visualizar-page');

}

async function generateReports() {

relatoriosContainer.innerHTML = '<p class="text-center text-gray-500">Gerando relatórios...</p>';

try {

const querySnapshot = await getDocs(historicoCollectionRef);

const historico = [];

querySnapshot.forEach(doc => historico.push(doc.data()));

if (historico.length === 0) {

relatoriosContainer.innerHTML = '<p class="text-center text-gray-500">Não há dados suficientes no histórico para gerar relatórios.</p>';

return;

}

const atendimentosPorProfissional = historico.reduce((acc, curr) => {

const profNome = curr.professional.nome;

acc[profNome] = (acc[profNome] || 0) + 1;

return acc;

}, {});

const entradasPorProfissional = historico.reduce((acc, curr) => {

const recepcaoNome = curr.acolhimento.profissionalNome;

acc[recepcaoNome] = (acc[recepcaoNome] || 0) + 1;

return acc;

}, {});

relatoriosContainer.innerHTML = `

<div class="bg-gray-100 p-6 rounded-xl shadow-inner">

<h3 class="text-xl font-bold text-gray-800 mb-4">Atendimentos por Profissional</h3>

<ul class="space-y-2">

${Object.entries(atendimentosPorProfissional).map(([nome, count]) => `

<li class="text-gray-700"><span class="font-semibold">${nome}:</span> ${count} atendimento(s)</li>

`).join('')}

</ul>

</div>

<div class="bg-gray-100 p-6 rounded-xl shadow-inner">

<h3 class="text-xl font-bold text-gray-800 mb-4">Entradas de Pacientes por Profissional de Acolhimento</h3>

<ul class="space-y-2">

${Object.entries(entradasPorProfissional).map(([nome, count]) => `

<li class="text-gray-700"><span class="font-semibold">${nome}:</span> ${count} entrada(s)</li>

`).join('')}

</ul>

</div>

`;

} catch (error) {

console.error("Erro ao gerar relatórios:", error);

relatoriosContainer.innerHTML = '<p class="text-center text-red-500">Erro ao gerar relatórios.</p>';

}

}

// Funções para a integração com a API Gemini

async function analyzeSymptoms(symptoms) {

aiAnalysisContainer.classList.remove('hidden');

aiAnalysisText.textContent = 'Analisando os sintomas... Por favor, aguarde.';

const maxRetries = 5;

let retryCount = 0;

let success = false;

let resultText = '';

while (retryCount < maxRetries && !success) {

try {

const prompt = `Analise a seguinte descrição de queixas e sintomas de um paciente e forneça uma breve análise em português (Brasil) para um profissional de saúde. Use um formato de lista para facilitar a leitura. Não inclua informações falsas ou diagnósticos definitivos.

Queixas e sintomas: ${symptoms}`;

const payload = {

contents: [{ role: "user", parts: [{ text: prompt }] }]

};

const apiKey = "";

const apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash-preview-05-20:generateContent?key=${apiKey}`;

const response = await fetch(apiUrl, {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify(payload)

});

if (!response.ok) {

throw new Error(`HTTP error! status: ${response.status}`);

}

const result = await response.json();

if (result.candidates && result.candidates.length > 0 &&

result.candidates[0].content && result.candidates[0].content.parts &&

result.candidates[0].content.parts.length > 0) {

resultText = result.candidates[0].content.parts[0].text;

success = true;

} else {

throw new Error('Resposta inesperada da API.');

}

} catch (error) {

console.error("Erro na chamada da API Gemini:", error);

retryCount++;

const delay = Math.pow(2, retryCount) \* 1000;

if (retryCount < maxRetries) {

console.log(`Tentando novamente em ${delay / 1000} segundos...`);

await new Promise(res => setTimeout(res, delay));

}

}

}

if (success) {

aiAnalysis = resultText;

aiAnalysisText.textContent = aiAnalysis;

} else {

aiAnalysisText.textContent = 'Não foi possível gerar a análise. Por favor, tente novamente mais tarde.';

aiAnalysis = '';

}

}

// --- Eventos ---

window.onload = async () => {

await initializeFirebase();

}

loginForm.addEventListener('submit', async (event) => {

event.preventDefault();

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

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

try {

const querySnapshot = await getDocs(usersCollectionRef);

const users = {};

querySnapshot.forEach(doc => {

users[doc.id] = doc.data();

});

if (users[username] && users[username].password === password) {

const userData = users[username];

currentUser = username;

currentUserRole = userData.role;

errorMessage.classList.add('hidden');

sidebar.classList.remove('hidden');

acompanhamentoBtn.classList.remove('hidden');

logoutBtn.classList.remove('hidden');

if (currentUserRole === 'recepcao') {

addNewPatientBtn.classList.remove('hidden');

} else if (currentUserRole === 'tecnico') {

historicoBtn.classList.remove('hidden');

} else if (currentUserRole === 'enfermeiro' || currentUserRole === 'medico') {

historicoBtn.classList.remove('hidden');

} else if (currentUserRole === 'gerente') {

historicoBtn.classList.remove('hidden');

relatoriosBtn.classList.remove('hidden');

gerenciarBtn.classList.remove('hidden');

}

showPage('acompanhamento-page');

} else {

errorMessage.classList.remove('hidden');

}

} catch (error) {

console.error("Erro ao fazer login:", error);

errorMessage.textContent = "Erro ao conectar ao banco de dados. Tente novamente.";

errorMessage.classList.remove('hidden');

}

});

patientForm.addEventListener('submit', async (event) => {

event.preventDefault();

const now = new Date();

const novoPaciente = {

nome: document.getElementById('nome').value,

horaChegada: now.toLocaleTimeString('pt-BR'),

status: 'Aguardando Pré-consulta',

classificacaoRisco: 'Neutro',

acolhimento: {

nome: document.getElementById('nome').value,

cartaoSus: document.getElementById('cartaoSus').value,

cpf: document.getElementById('cpf').value,

contato: document.getElementById('contato').value,

dataNascimento: document.getElementById('dataNascimento').value,

agenteSaude: document.getElementById('agenteSaude').value,

equipe: document.getElementById('equipe').value,

profissionalNome: currentUser,

timestamp: now.toLocaleString('pt-BR')

}

};

try {

await addDoc(pacientesCollectionRef, novoPaciente);

patientForm.reset();

showPage('acompanhamento-page');

} catch (error) {

console.error("Erro ao adicionar paciente:", error);

// Exibir mensagem de erro para o usuário

}

});

preConsultaForm.addEventListener('submit', async (event) => {

event.preventDefault();

const pacienteRef = doc(db, 'artifacts', appId, 'public', 'data', 'pacientes', pacienteSelecionadoId);

const updateData = {

status: 'Aguardando Profissional',

classificacaoRisco: document.getElementById('classificacaoRisco').value,

preConsulta: {

hipertensao: document.getElementById('hipertensao').value,

diabetes: document.getElementById('diabetes').value,

alergias: document.getElementById('alergias').value,

situacaoEspecial: document.getElementById('situacaoEspecial').value,

classificacaoRisco: document.getElementById('classificacaoRisco').value,

frequencia: document.getElementById('frequencia').value,

peso: document.getElementById('peso').value,

altura: document.getElementById('altura').value,

pressao: document.getElementById('pressao').value,

temperatura: document.getElementById('temperatura').value,

queixas: document.getElementById('queixas').value,

aiAnalysis: aiAnalysis,

profissionalNome: currentUser,

timestamp: new Date().toLocaleString('pt-BR')

}

};

try {

await updateDoc(pacienteRef, updateData);

preConsultaForm.reset();

showPage('acompanhamento-page');

} catch (error) {

console.error("Erro ao atualizar pré-consulta:", error);

}

});

professionalForm.addEventListener('submit', async (event) => {

event.preventDefault();

const encaminhamento = event.submitter.dataset.encaminhamento;

const pacienteRef = doc(db, 'artifacts', appId, 'public', 'data', 'pacientes', pacienteSelecionadoId);

try {

// Recuperar o paciente antes de apagar para ter todos os dados

const pacienteDoc = await getDoc(pacienteRef);

const pacienteData = pacienteDoc.data();

const profissionalData = {

nome: currentUser,

registro: currentUserRole,

observacoes: document.getElementById('prof-observacoes').value,

encaminhamento: encaminhamento,

timestamp: new Date().toLocaleString('pt-BR')

};

const historicoData = {

...pacienteData,

status: 'Atendimento Finalizado',

professional: profissionalData

};

await addDoc(historicoCollectionRef, historicoData);

await deleteDoc(pacienteRef);

professionalForm.reset();

showPage('acompanhamento-page');

} catch (error) {

console.error("Erro ao finalizar atendimento:", error);

}

});

createUserForm.addEventListener('submit', async (event) => {

event.preventDefault();

const newUsername = document.getElementById('new-username').value;

const newPassword = document.getElementById('new-password').value;

const newRole = document.getElementById('new-role').value;

try {

const userDocRef = doc(usersCollectionRef, newUsername);

const userSnapshot = await getDoc(userDocRef);

if (userSnapshot.exists()) {

userCreationMessage.textContent = 'Este nome de usuário já existe.';

userCreationMessage.className = 'text-red-500 text-sm text-center mt-2';

} else {

await setDoc(userDocRef, { password: newPassword, role: newRole });

userCreationMessage.textContent = `Usuário '${newUsername}' (${newRole}) criado com sucesso!`;

userCreationMessage.className = 'text-green-500 text-sm text-center mt-2';

createUserForm.reset();

}

} catch (error) {

console.error("Erro ao criar usuário:", error);

userCreationMessage.textContent = "Erro ao criar usuário. Tente novamente.";

userCreationMessage.className = 'text-red-500 text-sm text-center mt-2';

}

});

analyzeSymptomsBtn.addEventListener('click', () => {

const queixas = queixasTextarea.value.trim();

if (queixas) {

analyzeSymptoms(queixas);

} else {

aiAnalysisText.textContent = 'Por favor, insira as queixas do paciente para iniciar a análise.';

aiAnalysisContainer.classList.remove('hidden');

}

});

// Listeners de navegação

acompanhamentoBtn.addEventListener('click', () => { showPage('acompanhamento-page'); });

historicoBtn.addEventListener('click', () => { showPage('historico-page'); });

relatoriosBtn.addEventListener('click', () => { generateReports(); showPage('relatorios-page'); });

gerenciarBtn.addEventListener('click', () => { showPage('gerenciar-page'); });

logoutBtn.addEventListener('click', () => resetApp());

historicoBackBtn.addEventListener('click', () => showPage('acompanhamento-page'));

relatoriosBackBtn.addEventListener('click', () => showPage('acompanhamento-page'));

visualizarBackBtn.addEventListener('click', () => {

if (currentUserRole === 'recepcao' || currentUserRole === 'tecnico' || ['medico', 'enfermeiro'].includes(currentUserRole)) {

showPage('acompanhamento-page');

} else {

showPage('historico-page');

}

});

gerenciarBackBtn.addEventListener('click', () => showPage('acompanhamento-page'));

addNewPatientBtn.addEventListener('click', () => {

patientForm.reset();

showPage('acolhimento-page');

});

</script>

</body>

</html>