**<?php**

// api.php

require\_once 'config.php';

header('Content-Type: application/json; charset=utf-8');

if (empty($\_SESSION['user'])) {

    echo json\_encode(['ok'=>false,'msg'=>'Não autenticado']); exit;

}

$me = $\_SESSION['user'];

$mysqli = db\_connect();

$action = $\_REQUEST['action'] ?? '';

function json\_die($arr){ echo json\_encode($arr); exit; }

switch ($action) {

  case 'add\_visitor':

    // recepção adiciona visitante

    if ($me['role'] !== 'recep' && $me['role'] !== 'admin') json\_die(['ok'=>false,'msg'=>'Acesso negado']);

    $name = trim($\_POST['name'] ?? '');

    $doc = trim($\_POST['doc'] ?? '');

    $council = trim($\_POST['council'] ?? '');

    $reason = trim($\_POST['reason'] ?? '');

    if ($name === '' || $council === '' || $reason === '') json\_die(['ok'=>false,'msg'=>'Campos obrigatórios']);

    $stmt = $mysqli->prepare("INSERT INTO visitors (name, doc, council, reason, added\_by) VALUES (?, ?, ?, ?, ?)");

    $stmt->bind\_param('sssss', $name, $doc, $council, $reason, $me['username']);

    $ok = $stmt->execute();

    $stmt->close();

    json\_die(['ok'=>$ok, 'id'=>$mysqli->insert\_id]);

    break;

  case 'list\_visitors':

    // lista visitantes (opcional filters)

    $filter = $\_GET['filter'] ?? 'all'; // pending/present/all

    $from = $\_GET['from'] ?? null;

    $to = $\_GET['to'] ?? null;

    $userFilter = $\_GET['user'] ?? 'all';

    $sql = "SELECT \* FROM visitors WHERE 1=1";

    $params = [];

    $types = '';

    if ($filter === 'pending') { $sql .= " AND status='waiting'"; }

    elseif ($filter === 'present') { $sql .= " AND status='inside'"; }

    if ($userFilter !== 'all') { $sql .= " AND (added\_by = ?)"; $types .= 's'; $params[] = $userFilter; }

    if ($from) { $sql .= " AND added\_at >= ?"; $types .= 's'; $params[] = $from . " 00:00:00"; }

    if ($to) { $sql .= " AND added\_at <= ?"; $types .= 's'; $params[] = $to . " 23:59:59"; }

    $sql .= " ORDER BY added\_at DESC LIMIT 1000";

    if ($stmt = $mysqli->prepare($sql)) {

        if ($types !== '') $stmt->bind\_param($types, ...$params);

        $stmt->execute();

        $res = $stmt->get\_result();

        $rows = $res->fetch\_all(MYSQLI\_ASSOC);

        $stmt->close();

        json\_die(['ok'=>true,'data'=>$rows]);

    } else {

        json\_die(['ok'=>false,'msg'=>$mysqli->error]);

    }

    break;

  case 'approve':

    // aprovar entrada (recepção ou vereador)

    $id = intval($\_POST['id'] ?? 0);

    if ($id <= 0) json\_die(['ok'=>false,'msg'=>'ID inválido']);

    // se for vereador, só pode aprovar visitantes destinados ao seu name

    // BUT we stored council as name; so we check that when role=vereador

    if ($me['role'] === 'vereador') {

      $stmt = $mysqli->prepare("SELECT council FROM visitors WHERE id = ?");

      $stmt->bind\_param('i', $id); $stmt->execute(); $r = $stmt->get\_result()->fetch\_assoc(); $stmt->close();

      if (!$r || $r['council'] !== $me['fullName']) json\_die(['ok'=>false,'msg'=>'Não autorizado para aprovar este visitante']);

    }

    $now = date('Y-m-d H:i:s');

    $stmt = $mysqli->prepare("UPDATE visitors SET status='inside', entered\_at = ?, approved\_by = ? WHERE id = ?");

    $stmt->bind\_param('ssi', $now, $me['username'], $id);

    $ok = $stmt->execute();

    $stmt->close();

    json\_die(['ok'=>$ok]);

    break;

    case 'deny':

    // negar visita (recepção ou vereador)

    $id = intval($\_POST['id'] ?? 0);

    if ($id <= 0) json\_die(['ok'=>false,'msg'=>'ID inválido']);

    // se for vereador, só pode negar visitantes destinados ao seu fullName

    if ($me['role'] === 'vereador') {

        $stmt = $mysqli->prepare("SELECT council, status FROM visitors WHERE id = ?");

        $stmt->bind\_param('i', $id); $stmt->execute();

        $row = $stmt->get\_result()->fetch\_assoc();

        $stmt->close();

        if (!$row) json\_die(['ok'=>false,'msg'=>'Registro não encontrado']);

        if ($row['council'] !== $me['fullName']) json\_die(['ok'=>false,'msg'=>'Não autorizado para negar este visitante']);

        // opcional: não negar se já estiver inside/left

        if ($row['status'] !== 'waiting') json\_die(['ok'=>false,'msg'=>'Somente visitas em espera podem ser negadas']);

    }

    // quem pode negar: admin, recep, ou vereador responsável (já checado)

    if (!($me['role'] === 'admin' || $me['role'] === 'recep' || $me['role'] === 'vereador')) {

        json\_die(['ok'=>false,'msg'=>'Sem permissão para negar visite']);

    }

    $now = date('Y-m-d H:i:s');

    // Observação: sua tabela status tem 'waiting','inside','left' —

    // aqui definimos 'left' (como "não entrou") e guardamos exited\_by para registro

    $stmt = $mysqli->prepare("UPDATE visitors SET status = 'left', left\_at = ?, exited\_by = ? WHERE id = ?");

    $stmt->bind\_param('ssi', $now, $me['username'], $id);

    $ok = $stmt->execute();

    $stmt->close();

    json\_die(['ok'=>$ok]);

    break;

  case 'exit':

    $id = intval($\_POST['id'] ?? 0);

    if ($id <= 0) json\_die(['ok'=>false,'msg'=>'ID inválido']);

    // somente quem aprovou OU vereador responsável pode registrar saida (simple rule)

    $stmt = $mysqli->prepare("SELECT approved\_by, council FROM visitors WHERE id = ?");

    $stmt->bind\_param('i',$id); $stmt->execute(); $r = $stmt->get\_result()->fetch\_assoc(); $stmt->close();

    if (!$r) json\_die(['ok'=>false,'msg'=>'Registro não encontrado']);

    // autorizações: admin, recep, or if vereador and council==me.fullName OR if approved\_by==me.username

    if (!($me['role']==='admin' || $me['role']==='recep' || ($me['role']==='vereador' && $r['council']===$me['fullName']) || $r['approved\_by']===$me['username'])) {

      json\_die(['ok'=>false,'msg'=>'Sem permissão para registrar saída']);

    }

    $now = date('Y-m-d H:i:s');

    $stmt = $mysqli->prepare("UPDATE visitors SET status='left', left\_at = ?, exited\_by = ? WHERE id = ?");

    $stmt->bind\_param('ssi', $now, $me['username'], $id);

    $ok = $stmt->execute();

    $stmt->close();

    json\_die(['ok'=>$ok]);

    break;

  case 'delete\_filtered':

    // apagar apenas os registros que estão sendo exibidos/filtreados

    if ($me['role'] !== 'recep' && $me['role'] !== 'admin') json\_die(['ok'=>false,'msg'=>'Acesso negado']);

    // Accept filters via POST: user, from, to, council (optional)

    $userFilter = $\_POST['user'] ?? 'all';

    $from = $\_POST['from'] ?? null;

    $to = $\_POST['to'] ?? null;

    $council = $\_POST['council'] ?? null;

    $sql = "DELETE FROM visitors WHERE 1=1";

    $types = ''; $params=[];

    if ($userFilter !== 'all') { $sql .= " AND added\_by = ?"; $types.='s'; $params[] = $userFilter; }

    if ($from) { $sql .= " AND added\_at >= ?"; $types.='s'; $params[] = $from . " 00:00:00"; }

    if ($to) { $sql .= " AND added\_at <= ?"; $types.='s'; $params[] = $to . " 23:59:59"; }

    if ($council) { $sql .= " AND council = ?"; $types.='s'; $params[] = $council; }

    if ($stmt = $mysqli->prepare($sql)) {

      if ($types !== '') $stmt->bind\_param($types, ...$params);

      $ok = $stmt->execute();

      $affected = $stmt->affected\_rows;

      $stmt->close();

      json\_die(['ok'=>true,'deleted'=>$affected]);

    } else {

      json\_die(['ok'=>false,'msg'=>$mysqli->error]);

    }

    break;

  case 'get\_users':

    // lista usuários para filtro

    $res = $mysqli->query("SELECT username, fullName FROM users ORDER BY fullName");

    $arr = $res->fetch\_all(MYSQLI\_ASSOC);

    json\_die(['ok'=>true,'data'=>$arr]);

    break;

  case 'my\_pending\_for\_vereador':

    // para a tela do vereador — retorna waiting visitors whose council == my fullName

    $stmt = $mysqli->prepare("SELECT \* FROM visitors WHERE status='waiting' AND council = ? ORDER BY added\_at DESC");

    $stmt->bind\_param('s', $me['fullName']);

    $stmt->execute();

    $rows = $stmt->get\_result()->fetch\_all(MYSQLI\_ASSOC);

    $stmt->close();

    json\_die(['ok'=>true,'data'=>$rows]);

    break;

  case 'my\_present\_for\_vereador':

    $stmt = $mysqli->prepare("SELECT \* FROM visitors WHERE status='inside' AND council = ? ORDER BY entered\_at DESC");

    $stmt->bind\_param('s', $me['fullName']);

    $stmt->execute();

    $rows = $stmt->get\_result()->fetch\_all(MYSQLI\_ASSOC);

    $stmt->close();

    json\_die(['ok'=>true,'data'=>$rows]);

    break;

  default:

    json\_die(['ok'=>false,'msg'=>'Ação desconhecida']);

}

// chat.js

// polling chat basic

const API = 'chat\_api.php';

let pollTimer = null;

let sidebarTimer = null;

let audioPing = null;

// load audio

try {

  audioPing = new Audio();

  audioPing.src = 'data:audio/mp3;base64,//uQx...'; // optional: placeholder (you can replace with real file path 'ping.mp3')

} catch(e){ audioPing = null; }

function fetchContacts(){

  fetch(API + '?action=list\_contacts').then(r=>r.json()).then(j=>{

    if(!j.ok) return;

    const box = document.getElementById('contactsList');

    box.innerHTML = '';

    j.data.forEach(c=>{

      const el = document.createElement('div');

      el.className = 'contact';

      el.dataset.username = c.username;

      el.innerHTML = `<div>

          <div class="name">${escapeHtml(c.fullName||c.username)}</div>

          <div class="preview small-muted">${escapeHtml(c.role||'')}</div>

        </div>

        <div><span class="badge" id="badge-${c.username}" style="display:none">0</span></div>`;

      el.addEventListener('click', ()=> openConversation(c.username, c.fullName));

      box.appendChild(el);

    });

  });

}

function openConversation(username, full){

  currentWith = username;

  document.getElementById('convWith').textContent = full || username;

  document.getElementById('convSub').textContent = 'Carregando...';

  document.getElementById('msgText').disabled = false;

  document.getElementById('sendBtn').disabled = false;

  // highlight in sidebar

  document.querySelectorAll('.contact').forEach(el=>{

    el.classList.toggle('contact-selected', el.dataset.username===username);

  });

  loadConversation(username);

}

function loadConversation(username){

  fetch(API + '?action=get\_conversation&with=' + encodeURIComponent(username)).then(r=>r.json()).then(j=>{

    if(!j.ok) return;

    renderMessages(j.data);

    // clear badge

    const b = document.getElementById('badge-' + username);

    if (b) b.style.display='none';

  });

}

function renderMessages(msgs){

  const area = document.getElementById('messagesArea');

  area.innerHTML = '';

  if (!msgs.length) {

    area.innerHTML = '<div class="empty-center">Sem mensagens</div>';

    return;

  }

  msgs.forEach(m=>{

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

    div.className = 'msg ' + (m.sender === me ? 'me' : 'other');

    div.dataset.id = m.id;

    div.innerHTML = `<div>${escapeHtml(m.message)}</div>

      <div class="meta">${escapeHtml(m.sender)} • ${m.created\_at}

        <button style="margin-left:8px;font-size:12px;border:none;background:transparent;cursor:pointer;color:rgba(0,0,0,0.35)" onclick="deleteMessage(${m.id})">Apagar</button>

      </div>`;

    area.appendChild(div);

  });

  area.scrollTop = area.scrollHeight;

}

document.getElementById && document.addEventListener('DOMContentLoaded', ()=>{

  fetchContacts();

  document.getElementById('refreshContacts').addEventListener('click', fetchContacts);

  document.getElementById('sendBtn').addEventListener('click', sendMessage);

  document.getElementById('msgText').addEventListener('keydown', (e)=>{

    if(e.key === 'Enter' && !e.shiftKey){ e.preventDefault(); sendMessage(); }

  });

  // poll unread / sidebar every 4s

  sidebarTimer = setInterval(pollUnread, 4000);

  pollUnread();

});

function pollUnread(){

  fetch(API + '?action=poll\_unread').then(r=>r.json()).then(j=>{

    if(!j.ok) return;

    // show global notification in top? Here we update badges for contacts via recent\_preview

    fetch(API + '?action=recent\_preview').then(r=>r.json()).then(resp=>{

      if(!resp.ok) return;

      // resp.data: other,message,created\_at,from

      resp.data.forEach(item=>{

        const other = item.other;

        const badge = document.getElementById('badge-' + other);

        if (!badge) return;

        // if message from other to me and not read -> show badge

        if (item.from !== me) {

          badge.textContent = '1';

          badge.style.display = 'inline-block';

          // play sound if conversation not open

          if (currentWith !== other) {

            if (audioPing) { try { audioPing.currentTime = 0; audioPing.play(); } catch(e){} }

          }

        } else {

          badge.style.display = 'none';

        }

      });

    });

  });

}

function sendMessage(){

  if (!currentWith) return alert('Selecione um contato.');

  const ta = document.getElementById('msgText');

  const txt = ta.value.trim();

  if (!txt) return;

  const form = new URLSearchParams();

  form.append('action','send');

  form.append('to', currentWith);

  form.append('message', txt);

  fetch(API, {method:'POST', body: form}).then(r=>r.json()).then(j=>{

    if (j.ok) {

      ta.value = '';

      loadConversation(currentWith);

    } else {

      alert(j.msg || 'Erro ao enviar');

    }

  });

}

function deleteMessage(id){

  if (!confirm('Apagar essa mensagem?')) return;

  const f = new URLSearchParams();

  f.append('action','delete\_message');

  f.append('id', id);

  fetch(API, {method:'POST', body: f}).then(r=>r.json()).then(j=>{

    if (j.ok) {

      // remove element

      const el = document.querySelector(`.msg[data-id="${id}"]`);

      if (el) el.remove();

    } else alert(j.msg || 'Erro ao apagar');

  });

}

function escapeHtml(s){ if (s===null||s===undefined) return ''; return String(s).replace(/[&<>"']/g,m=>({'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;',"'":'&#39;'}[m])); }

**<?php**

// chat.php

require\_once 'config.php';

if (empty($\_SESSION['user'])) {

    header('Location: index.php');

    exit;

}

$user = $\_SESSION['user']; // tem username, fullName, role

$username = htmlspecialchars($user['username']);

$fullName = htmlspecialchars($user['fullName'] ?? $user['username']);

**?>**

<!doctype html>

<html lang="pt-BR">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <title>Chat Interno - Recepção</title>

  <link rel="stylesheet" href="style.css">

  <style>

  /\* Chat minimal — mantém a identidade visual (cores, sombras) \*/

  .chat-wrap { max-width:1100px; margin:24px auto; display:flex; gap:18px; align-items:flex-start; padding:18px; }

  .chat-list { width:300px; background:var(--card); border-radius:12px; box-shadow:var(--shadow); padding:12px; overflow:auto; max-height:70vh; }

  .chat-list .contact { padding:10px; border-radius:8px; display:flex; justify-content:space-between; gap:8px; cursor:pointer; align-items:center; }

  .chat-list .contact:hover { background:#f0f7ff; transform:translateX(4px); }

  .chat-list .contact.active { background: linear-gradient(90deg,var(--accent),var(--accent-700)); color:#fff; }

  .chat-main { flex:1; display:flex; flex-direction:column; background:var(--card); border-radius:12px; box-shadow:var(--shadow); max-height:80vh; overflow:hidden; }

  .chat-header { padding:14px; border-bottom:1px solid #eef3f9; display:flex; align-items:center; gap:12px; }

  .chat-body { padding:16px; overflow:auto; flex:1; background:linear-gradient(#fbfdff,#f7fbff); }

  .chat-footer { padding:12px; border-top:1px solid #eef3f9; display:flex; gap:8px; align-items:center; }

  .msg { max-width:75%; margin-bottom:10px; padding:10px 12px; border-radius:12px; box-shadow:0 6px 18px rgba(11,18,28,0.06); }

  .msg.me { margin-left:auto; background:linear-gradient(90deg,var(--accent),var(--accent-700)); color:#fff; border-bottom-right-radius:4px; }

  .msg.them { margin-right:auto; background:#fff; color:#111; border-bottom-left-radius:4px; }

  .msg .time{ display:block; font-size:12px; color:var(--muted); margin-top:6px; opacity:0.9 }

  .input { flex:1; padding:10px 12px; border-radius:10px; border:2px solid #e6edf3; outline:none; }

  .btn-send { padding:10px 14px; border-radius:10px; border:none; background:linear-gradient(90deg,var(--accent),var(--accent-700)); color:#fff; font-weight:700; cursor:pointer; }

  .unread-badge { background:#ef4444;color:white;padding:4px 8px;border-radius:999px;font-weight:700;font-size:12px; }

  /\* responsivo \*/

  @media (max-width:900px){

    .chat-wrap { flex-direction:column; }

    .chat-list { width:100%; max-height:200px; order:2; }

    .chat-main { order:1; max-height:60vh; }

  }

  </style>

</head>

<body class="panel-page">

  <header class="topbar light">

    <div class="brand">

      <img src="https://tse3.mm.bing.net/th/id/OIP.NeJbj2QckKlAAfZ0YlkgUgHaJw?cb=12&pid=Api" alt="Brasão" class="brasao">

      <div>

        <div class="title">Chat Interno</div>

        <div class="subtitle">Converse com vereadores e recepcionistas</div>

      </div>

    </div>

    <div class="top-actions">

      <div id="clock" class="clock">--:--:--</div>

      <div class="user-display">**<?php** echo $fullName;**?>**</div>

      <button class="btn ghost" onclick="location.href='painel.php'">Voltar</button>

    </div>

  </header>

  <main class="chat-wrap">

    <aside class="chat-list" id="contacts">

      <!-- contatos carregados por JS -->

      <div style="font-weight:700;margin-bottom:8px">Contatos</div>

      <div id="contactsList">Carregando...</div>

    </aside>

    <section class="chat-main">

      <div class="chat-header" id="chatHeader">

        <div style="font-weight:800" id="chatWith">Selecione um contato</div>

        <div style="margin-left:auto" id="contactBadge"></div>

      </div>

      <div class="chat-body" id="chatBody">Selecione um contato para começar a conversar.</div>

      <div class="chat-footer">

        <input id="chatInput" class="input" placeholder="Escreva uma mensagem..." disabled>

        <button id="btnSend" class="btn-send" disabled>Enviar</button>

      </div>

    </section>

  </main>

  <div id="toast-container" class="toast-container"></div>

  <!-- Som de notificação (arquivo embutido como base64 simples Opcional) -->

  <audio id="pingAudio" src=""></audio>

<script>

const myUser = '**<?php** echo addslashes($username); **?>**';

let currentContact = null;

let pollInterval = null;

let lastMessageId = 0;

// helper

function el(q){ return document.querySelector(q); }

function escapeHtml(s){ return String(s||'').replace(/[&<>"']/g,m=>({'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;',"'":'&#39;'}[m])); }

function showToast(text, type='success'){

  const c = document.getElementById('toast-container');

  if(!c) return;

  const t = document.createElement('div');

  t.className = 'toast ' + (type==='error'?'error':'success');

  t.innerHTML = '<strong>'+escapeHtml(text)+'</strong>';

  c.appendChild(t);

  setTimeout(()=>{ t.style.opacity='0'; setTimeout(()=>t.remove(),400); }, 3200);

}

// carrega lista de contatos (quem trocou msgs ou usuários do sistema)

async function loadContacts(){

  const res = await fetch('chat\_api.php?action=list\_contacts');

  const j = await res.json();

  const list = document.getElementById('contactsList');

  list.innerHTML = '';

  if(!j.ok){ list.textContent = 'Erro ao carregar'; return; }

  j.data.forEach(c=>{

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

    div.className = 'contact';

    if(c.username === currentContact) div.classList.add('active');

    div.dataset.username = c.username;

    div.innerHTML = `<div><strong>${escapeHtml(c.fullName)}</strong><div style="font-size:13px;color:var(--muted)">${escapeHtml(c.username)}</div></div><div>${c.unread? '<span class="unread-badge">'+c.unread+'</span>':''}</div>`;

    div.addEventListener('click', ()=> openChatWith(c.username, c.fullName));

    list.appendChild(div);

  });

}

// abre conversa com contato

function openChatWith(username, fullName){

  currentContact = username;

  el('#chatWith').textContent = fullName + ' ('+username+')';

  el('#chatInput').disabled = false;

  el('#btnSend').disabled = false;

  lastMessageId = 0;

  el('#chatBody').innerHTML = 'Carregando...';

  markRead(username);

  loadMessages();

  if(pollInterval) clearInterval(pollInterval);

  pollInterval = setInterval(()=> { loadMessages(); loadContacts(); }, 2000);

  loadContacts();

}

// marca mensagens como lidas (server-side)

async function markRead(contact){

  await fetch('chat\_api.php?action=mark\_read', {

    method:'POST', headers:{'Content-Type':'application/x-www-form-urlencoded'}, body:'contact='+encodeURIComponent(contact)

  });

  loadContacts();

}

// carrega mensagens

async function loadMessages(){

  if(!currentContact) return;

  const res = await fetch('chat\_api.php?action=fetch&contact='+encodeURIComponent(currentContact)+'&since\_id='+lastMessageId);

  const j = await res.json();

  if(!j.ok) return;

  const body = el('#chatBody');

  // if full replace when since\_id==0

  if(Number(j.since\_id) === 0 || lastMessageId === 0){

    body.innerHTML = '';

  }

  j.data.forEach(m=>{

    lastMessageId = Math.max(lastMessageId, m.id);

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

    div.className = 'msg ' + (m.sender === myUser ? 'me' : 'them');

    div.innerHTML = `<div>${escapeHtml(m.message)}</div><span class="time">${escapeHtml(m.sender)} • ${escapeHtml(m.sent\_at)}</span>`;

    body.appendChild(div);

  });

  body.scrollTop = body.scrollHeight;

  if(j.playPing) {

    const audio = document.getElementById('pingAudio');

    if(audio && audio.src) audio.play().catch(()=>{});

  }

}

// enviar mensagem

async function sendMessage(){

  const txt = el('#chatInput').value.trim();

  if(!txt || !currentContact) return;

  el('#chatInput').value = '';

  const res = await fetch('chat\_api.php?action=send', {

    method:'POST', headers:{'Content-Type':'application/x-www-form-urlencoded'},

    body: `receiver=${encodeURIComponent(currentContact)}&message=${encodeURIComponent(txt)}`

  });

  const j = await res.json();

  if(j.ok){

    loadMessages();

    loadContacts();

  } else {

    showToast(j.msg || 'Erro ao enviar', 'error');

  }

}

document.addEventListener('DOMContentLoaded', ()=>{

  loadContacts();

  document.getElementById('btnSend').addEventListener('click', sendMessage);

  document.getElementById('chatInput').addEventListener('keydown', (e)=>{ if(e.key === 'Enter' && !e.shiftKey) { e.preventDefault(); sendMessage(); }});

  // inicializa relógio (se tiver)

  const clockEl = document.getElementById('clock');

  if(clockEl) { function updateClock(){ clockEl.textContent = new Date().toLocaleTimeString(); } updateClock(); setInterval(updateClock,1000); }

});

</script>

</body>

</html>

**<?php**

// chat\_api.php

require\_once 'config.php';

header('Content-Type: application/json; charset=utf-8');

if (empty($\_SESSION['user'])) {

    echo json\_encode(['ok'=>false,'msg'=>'Não autenticado']);

    exit;

}

$me = $\_SESSION['user']['username'];

$action = $\_GET['action'] ?? $\_POST['action'] ?? '';

$mysqli = db\_connect();

function json\_ok($data=[]){ echo json\_encode(array\_merge(['ok'=>true], $data)); exit; }

function json\_err($msg='Erro'){ echo json\_encode(['ok'=>false,'msg'=>$msg]); exit; }

if ($action === 'list\_contacts') {

    // lista todos usuários do sistema, exceto o próprio

    $stmt = $mysqli->prepare("

        SELECT

            u.username,

            u.fullName,

            (SELECT COUNT(\*) FROM messages m

                WHERE m.receiver = ? AND m.sender = u.username

                AND m.read\_status = 'unread') AS unread

        FROM users u

        WHERE u.username != ?

        ORDER BY u.fullName ASC

    ");

    $stmt->bind\_param('ss', $me, $me);

    if(!$stmt->execute()) json\_err('DB erro');

    $res = $stmt->get\_result();

    $out = [];

    while($row = $res->fetch\_assoc()) {

        $out[] = [

            'username' => $row['username'],

            'fullName' => $row['fullName'] ?: $row['username'],

            'unread' => intval($row['unread'])

        ];

    }

    json\_ok(['data'=>$out]);

}

if ($action === 'fetch') {

    $contact = $\_GET['contact'] ?? '';

    $since\_id = intval($\_GET['since\_id'] ?? 0);

    if (!$contact) json\_err('Contato inválido');

    // fetch messages between me and contact newer than since\_id

    $stmt = $mysqli->prepare("SELECT id, sender, receiver, message, DATE\_FORMAT(sent\_at, '%d/%m/%Y %H:%i:%s') as sent\_at FROM messages WHERE ((sender = ? AND receiver = ?) OR (sender = ? AND receiver = ?)) AND id > ? ORDER BY id ASC");

    $stmt->bind\_param('ssssi', $me, $contact, $contact, $me, $since\_id);

    if(!$stmt->execute()) json\_err('DB fetch erro');

    $res = $stmt->get\_result();

    $arr = [];

    $playPing = false;

    while($r = $res->fetch\_assoc()){

      $arr[] = $r;

      // if message is from contact and unread -> we'll tell frontend to play ping

      if ($r['sender'] === $contact) $playPing = true;

    }

    json\_ok(['data'=>$arr, 'since\_id'=>$since\_id, 'playPing'=>$playPing]);

}

if ($action === 'send') {

    $receiver = $\_POST['receiver'] ?? '';

    $message = trim($\_POST['message'] ?? '');

    if (!$receiver || $message === '') json\_err('Dados inválidos');

    $stmt = $mysqli->prepare("INSERT INTO messages (sender, receiver, message, sent\_at, read\_status) VALUES (?, ?, ?, NOW(), 'unread')");

    $stmt->bind\_param('sss', $me, $receiver, $message);

    if(!$stmt->execute()) json\_err('Erro ao inserir');

    json\_ok(['id' => $mysqli->insert\_id]);

}

if ($action === 'mark\_read') {

    $contact = $\_POST['contact'] ?? '';

    if (!$contact) json\_err('Contato inválido');

    $stmt = $mysqli->prepare("UPDATE messages SET read\_status = 'read' WHERE sender = ? AND receiver = ? AND read\_status = 'unread'");

    $stmt->bind\_param('ss', $contact, $me);

    if(!$stmt->execute()) json\_err('Erro ao marcar lido');

    json\_ok(['updated' => $stmt->affected\_rows]);

}

echo json\_encode(['ok'=>false,'msg'=>'Ação desconhecida']);

exit;

**<?php**

// chat\_api.php

require\_once 'config.php';

header('Content-Type: application/json; charset=utf-8');

$mysqli = db\_connect();

session\_start();

if (empty($\_SESSION['user'])) {

    echo json\_encode(['ok'=>false,'msg'=>'Not logged']);

    exit;

}

$user = $\_SESSION['user'];

$me = $mysqli->real\_escape\_string($user['username']);

$action = isset($\_GET['action']) ? $\_GET['action'] : (isset($\_POST['action'])?$\_POST['action']:'');

function json($a){ echo json\_encode($a); exit; }

if ($action === 'list\_contacts') {

    // list all other users (simples)

    $res = $mysqli->query("SELECT username, fullName, role FROM users WHERE username != '{$me}'");

    $out = [];

    while($r = $res->fetch\_assoc()){

        $out[] = $r;

    }

    json(['ok'=>true,'data'=>$out]);

}

if ($action === 'get\_conversation') {

    $with = $mysqli->real\_escape\_string($\_GET['with'] ?? '');

    if (!$with) json(['ok'=>false,'msg'=>'missing with']);

    // select messages not deleted by the requesting party

    $sql = "SELECT id,sender,receiver,message,created\_at,is\_read FROM chat\_messages

            WHERE (sender='{$me}' AND receiver='{$with}' AND deleted\_by\_sender=0)

               OR (sender='{$with}' AND receiver='{$me}' AND deleted\_by\_receiver=0)

            ORDER BY created\_at ASC";

    $res = $mysqli->query($sql);

    $rows = [];

    while($r = $res->fetch\_assoc()) $rows[] = $r;

    // mark messages as read (those received by me)

    $mysqli->query("UPDATE chat\_messages SET is\_read=1 WHERE receiver='{$me}' AND sender='{$with}'");

    json(['ok'=>true,'data'=>$rows]);

}

if ($action === 'send') {

    $to = $mysqli->real\_escape\_string($\_POST['to'] ?? '');

    $msg = trim($\_POST['message'] ?? '');

    if (!$to || $msg === '') json(['ok'=>false,'msg'=>'invalid']);

    $stmt = $mysqli->prepare("INSERT INTO chat\_messages (sender,receiver,message) VALUES (?,?,?)");

    $stmt->bind\_param('sss',$me,$to,$msg);

    $ok = $stmt->execute();

    if (!$ok) json(['ok'=>false,'msg'=>$stmt->error]);

    json(['ok'=>true,'id'=>$mysqli->insert\_id]);

}

if ($action === 'poll\_unread') {

    // total unread for me

    $res = $mysqli->query("SELECT COUNT(\*) cnt FROM chat\_messages WHERE receiver='{$me}' AND is\_read=0 AND deleted\_by\_receiver=0");

    $cnt = $res->fetch\_assoc()['cnt'] ?? 0;

    json(['ok'=>true,'unread'=>intval($cnt)]);

}

if ($action === 'recent\_preview') {

    // return last message per contact for sidebar preview

    $sql = "SELECT m.\* FROM chat\_messages m

            JOIN (

              SELECT IF(sender='{$me}', receiver, sender) other, MAX(created\_at) mx

              FROM chat\_messages

              WHERE (sender='{$me}' OR receiver='{$me}')

                AND ((sender='{$me}' AND deleted\_by\_sender=0) OR (receiver='{$me}' AND deleted\_by\_receiver=0))

              GROUP BY other

            ) t ON ((m.sender='{$me}' AND m.receiver=t.other) OR (m.receiver='{$me}' AND m.sender=t.other)) AND m.created\_at = t.mx

            ORDER BY m.created\_at DESC

            LIMIT 50";

    $res = $mysqli->query($sql);

    $rows = [];

    while($r = $res->fetch\_assoc()) {

        $other = ($r['sender']===$me) ? $r['receiver'] : $r['sender'];

        $rows[] = ['other'=>$other,'message'=>$r['message'],'created\_at'=>$r['created\_at'],'from'=>$r['sender']];

    }

    json(['ok'=>true,'data'=>$rows]);

}

if ($action === 'delete\_message') {

    $id = intval($\_POST['id'] ?? 0);

    if (!$id) json(['ok'=>false,'msg'=>'missing id']);

    // check message exists and sender/receiver

    $res = $mysqli->query("SELECT sender,receiver FROM chat\_messages WHERE id={$id}");

    if (!$res || $res->num\_rows===0) json(['ok'=>false,'msg'=>'no message']);

    $row = $res->fetch\_assoc();

    if ($row['sender'] === $me) {

        $mysqli->query("UPDATE chat\_messages SET deleted\_by\_sender=1 WHERE id={$id}");

    } elseif ($row['receiver'] === $me) {

        $mysqli->query("UPDATE chat\_messages SET deleted\_by\_receiver=1 WHERE id={$id}");

    } else {

        json(['ok'=>false,'msg'=>'no permission']);

    }

    json(['ok'=>true]);

}

json(['ok'=>false,'msg'=>'unknown action']);

**<?php**

// chat\_ui.php

require\_once 'config.php';

session\_start();

if (empty($\_SESSION['user'])) { header('Location: index.php'); exit; }

$user = $\_SESSION['user'];

$me = htmlspecialchars($user['username']);

$myFull = htmlspecialchars($user['fullName']);

**?>**

<!doctype html>

<html lang="pt-BR">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1" />

<title>Chat — Painel</title>

<link rel="stylesheet" href="style.css"> <!-- usa seu css -->

<style>

/\* minimal styles for chat, matches seu visual \*/

.chat-wrap{max-width:1100px;margin:40px auto;display:flex;gap:18px}

.chat-sidebar{width:300px;background:var(--card);padding:12px;border-radius:14px;box-shadow:var(--shadow)}

.chat-main{flex:1;background:var(--card);padding:18px;border-radius:14px;box-shadow:var(--shadow);display:flex;flex-direction:column;min-height:520px}

.contact{display:flex;align-items:center;justify-content:space-between;padding:10px;border-radius:8px;cursor:pointer}

.contact:hover{background:#f3f8ff}

.contact .name{font-weight:700;color:var(--accent-700)}

.preview{font-size:13px;color:var(--muted);margin-top:4px}

.messages{flex:1;overflow:auto;padding:12px 8px}

.msg{max-width:70%;padding:10px;border-radius:10px;margin-bottom:8px;line-height:1.25}

.msg.me{background:linear-gradient(90deg,var(--accent),var(--accent-700));color:#fff;margin-left:auto;border-bottom-right-radius:2px}

.msg.other{background:#f1f6fb;color:#0b2740;border-bottom-left-radius:2px}

.msg .meta{font-size:11px;color:rgba(0,0,0,0.4);margin-top:6px}

.chat-input{display:flex;gap:8px;padding:10px;border-top:1px solid #eef3f9}

.chat-input textarea{flex:1;border-radius:10px;padding:10px;border:1px solid #e6edf3;height:56px;resize:none}

.btn-send{background:linear-gradient(90deg,var(--accent),var(--accent-700));color:#fff;border:none;padding:10px 16px;border-radius:10px;cursor:pointer}

.badge{background:#e53935;color:#fff;padding:4px 8px;border-radius:999px;font-weight:700}

.small-muted{font-size:13px;color:var(--muted)}

.empty-center{text-align:center;color:var(--muted);padding:30px}

.contact-selected{background:linear-gradient(90deg,#eef7ff,#f3f9ff);border-left:3px solid var(--accent);}

/\* mobile \*/

@media(max-width:900px){ .chat-wrap{flex-direction:column;padding:8px} .chat-sidebar{width:100%} .chat-main{min-height:420px} }

</style>

</head>

<body>

<header class="topbar light">

  <div class="brand">

    <img src="https://tse3.mm.bing.net/th/id/OIP.NeJbj2QckKlAAfZ0YlkgUgHaJw?cb=12&pid=Api" class="brasao" alt="">

    <div><div class="title">Chat Interno</div><div class="subtitle">Conversa entre recepção e vereadores</div></div>

  </div>

  <div class="top-actions">

    <div class="small-muted">Você: **<?php** echo $myFull; **?>**</div>

    <button class="btn ghost" onclick="location.href='painel.php'">Voltar</button>

  </div>

</header>

<main class="chat-wrap">

  <aside class="chat-sidebar">

    <div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:12px">

      <strong>Contatos</strong>

      <button class="btn ghost" id="refreshContacts">Atualizar</button>

    </div>

    <div id="contactsList"></div>

  </aside>

  <section class="chat-main">

    <div id="convHeader" style="padding-bottom:8px;border-bottom:1px solid #eef3f9">

      <strong id="convWith">Selecione um contato</strong>

      <div id="convSub" class="small-muted">Clique em um contato para iniciar</div>

    </div>

    <div class="messages" id="messagesArea">

      <div class="empty-center">Nenhuma conversa selecionada</div>

    </div>

    <div class="chat-input">

      <textarea id="msgText" placeholder="Escreva sua mensagem..." disabled></textarea>

      <button class="btn-send" id="sendBtn" disabled>Enviar</button>

    </div>

  </section>

</main>

<script>

const me = "**<?php** echo addslashes($user['username']); **?>**";

let currentWith = null;

</script>

<script src="chat.js"></script>

</body>

</html>

**<?php**

// config.php

define('DB\_HOST', 'localhost');

define('DB\_USER', 'fhiktajx\_adminuser');

define('DB\_PASS', 'AnalistaTI.2025');

define('DB\_NAME', 'fhiktajx\_painel\_recepcao'); // << banco correto

define('DB\_CHAR', 'utf8mb4');

function db\_connect(){

    $mysqli = new mysqli(DB\_HOST, DB\_USER, DB\_PASS, DB\_NAME);

    if ($mysqli->connect\_errno) {

        http\_response\_code(500);

        die("Erro ao conectar ao banco: " . $mysqli->connect\_error);

    }

    $mysqli->set\_charset(DB\_CHAR);

    return $mysqli;

}

session\_start();

**?>**

**<?php**

// create\_user.php - use somente temporariamente para criar usuários

require\_once 'config.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $u = trim($\_POST['username'] ?? '');

    $p = $\_POST['password'] ?? '';

    $full = trim($\_POST['fullName'] ?? $u);

    $role = $\_POST['role'] ?? 'recep';

    if (!$u || !$p) die('username/password required');

    $hash = password\_hash($p, PASSWORD\_DEFAULT);

    $db = db\_connect();

    $stmt = $db->prepare("INSERT INTO users (username, password\_hash, fullName, role) VALUES (?, ?, ?, ?)");

    $stmt->bind\_param('ssss', $u, $hash, $full, $role);

    $ok = $stmt->execute();

    echo $ok ? "Usuário criado" : "Erro: " . $stmt->error;

    exit;

}

**?>**

<form method="post">

  <input name="username" placeholder="username"><br>

  <input name="password" placeholder="password"><br>

  <input name="fullName" placeholder="Full name"><br>

  <select name="role"><option value="recep">recepção</option><option value="vereador">vereador</option><option value="admin">admin</option></select><br>

  <button type="submit">Criar</button>

</form>

//db.php

**<?php**

$DB\_HOST='localhost';$DB\_NAME='painel\_recepcao';$DB\_USER='usuario\_banco';$DB\_PASS='senha\_banco';

$mysqli=new mysqli($DB\_HOST,$DB\_USER,$DB\_PASS,$DB\_NAME);

if($mysqli->connect\_errno){die('Erro DB');}

**?>**

//gerar\_hash.php

**<?php**

$senha = 'Camara.2025'; // troca pela senha que quer usar

echo password\_hash($senha, PASSWORD\_DEFAULT);

**?>**

**<?php**

// index.php

// Se já logado, redireciona pro painel

require\_once 'config.php';

if (!empty($\_SESSION['user'])) {

    header('Location: painel.php');

    exit;

}

**?>**

<!doctype html>

<html lang="pt-BR">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <title>Login - Painel da Recepção</title>

  <link rel="stylesheet" href="style.css">

  <script defer src="script.js"></script>

</head>

<body class="login-page">

  <header class="top-header">

    <img src="https://tse3.mm.bing.net/th/id/OIP.NeJbj2QckKlAAfZ0YlkgUgHaJw?cb=12&pid=Api" alt="Brasão" class="brasao">

    <div class="header-title">Painel da Recepção – Câmara Municipal de Vitória de Santo Antão</div>

  </header>

  <main class="login-main">

    <div class="login-card">

      <div class="illustration">

        <div class="chars" id="chars">

          <div class="char char1"></div>

          <div class="char char2"></div>

          <div class="char char3"></div>

          <div class="char char4"></div>

        </div>

      </div>

      <div class="login-box">

        <h2>Bem-vindo</h2>

        <p class="muted">Faça login para acessar o painel de recepção</p>

        <form method="post" action="login.php" id="loginForm">

          <label>Usuário</label>

          <input id="username" name="username" type="text" placeholder="Usuário" autocomplete="off" required>

          <label>Senha</label>

          <input id="password" name="password" type="password" placeholder="Senha" required>

          <div class="hint muted"><strong></strong> <strong></strong></div>

          <button id="loginBtn" class="btn primary" type="submit">Entrar</button>

        </form>

        <p id="loginError" class="error"></p>

      </div>

    </div>

  </main>

  <div id="toast-container" class="toast-container"></div>

</body>

</html>

**<?php**

// keep\_alive.php

require\_once 'config.php';

header('Content-Type: application/json; charset=utf-8');

try {

    $mysqli = db\_connect();

    // Faz uma consulta simples pra "acordar" o banco

    $ping = $mysqli->query("SELECT NOW() as hora");

    if ($ping && $row = $ping->fetch\_assoc()) {

        echo json\_encode(['ok' => true, 'hora' => $row['hora']]);

    } else {

        echo json\_encode(['ok' => false, 'msg' => 'Ping falhou']);

    }

    $mysqli->close();

} catch (Exception $e) {

    echo json\_encode(['ok' => false, 'msg' => $e->getMessage()]);

}

**<?php**

// login.php

require\_once 'config.php';

if ($\_SERVER['REQUEST\_METHOD'] !== 'POST') {

    header('Location: index.php');

    exit;

}

$username = trim($\_POST['username'] ?? '');

$password = $\_POST['password'] ?? '';

if ($username === '' || $password === '') {

    $\_SESSION['login\_error'] = 'Preencha usuário e senha';

    header('Location: index.php');

    exit;

}

$mysqli = db\_connect();

$stmt = $mysqli->prepare("SELECT id, username, password\_hash, fullName, role FROM users WHERE username = ?");

$stmt->bind\_param('s', $username);

$stmt->execute();

$res = $stmt->get\_result();

$user = $res->fetch\_assoc();

$stmt->close();

if (!$user) {

    $\_SESSION['login\_error'] = 'Usuário não encontrado';

    header('Location: index.php');

    exit;

}

if (!password\_verify($password, $user['password\_hash'])) {

    $\_SESSION['login\_error'] = 'Senha incorreta';

    header('Location: index.php');

    exit;

}

// sucesso: grava sessão

$\_SESSION['user'] = [

    'id' => $user['id'],

    'username' => $user['username'],

    'fullName' => $user['fullName'],

    'role' => $user['role'],

];

header('Location: painel.php');

exit;

**<?php**

// logout.php

require\_once 'config.php';

session\_unset();

session\_destroy();

header('Location: index.php');

exit;

**<?php**

// painel.php

require\_once 'config.php';

if (empty($\_SESSION['user'])) {

    header('Location: index.php');

    exit;

}

$role = $\_SESSION['user']['role'] ?? 'recep';

if ($role === 'recep' || $role === 'admin') {

    header('Location: recepcao.php');

    exit;

} elseif ($role === 'vereador') {

    header('Location: vereador.php');

    exit;

} else {

    echo "Função desconhecida.";

    exit;

}

**<?php**

// recepcao.php

require\_once 'config.php';

if (empty($\_SESSION['user'])) {

    header('Location: index.php');

    exit;

}

$user = $\_SESSION['user'];

if ($user['role'] !== 'recep' && $user['role'] !== 'admin') {

    echo "Acesso negado.";

    exit;

}

// lista de vereadores (pode ser dinâmica, aqui você pode buscar de uma tabela; mantemos estática)

$councilors = [

  "Ana Paula", "André de Baú", "Baú Nogueira", "BIU De Genário",

  "Carlos Henrique Queiroz", "Celso Bezerra", "Denis Lima", "Diretoria",

  "Edmilson de Várzea Grande", "Everaldo Arruda", "Fabio Raylux", "Felipe Cezar",

  "Gold do Pneu", "Josias da Saúde", "JOTA Domingos", "Lourinaldo Júnior",

  "Mano Holanda", "Marcos da Prestação", "Mizael de Davi", "Novo da Banca", "TI"

];

**?>**

<!doctype html>

<html lang="pt-BR">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <title>Painel - Recepção</title>

  <link rel="stylesheet" href="style.css">

  <script defer src="script.js"></script>

</head>

<body class="panel-page">

  <header class="topbar light">

    <div class="brand">

      <img src="https://tse3.mm.bing.net/th/id/OIP.NeJbj2QckKlAAfZ0YlkgUgHaJw?cb=12&pid=Api" alt="Brasão" class="brasao">

      <div>

        <div class="title">Painel da Recepção</div>

        <div class="subtitle">Câmara Municipal de Vitória de Santo Antão</div>

      </div>

    </div>

    <div class="top-actions">

      <div id="clock" class="clock">--:--:--</div>

      <div id="userDisplay" class="user-display">**<?php** echo htmlspecialchars($user['fullName']); **?>**</div>

      <button id="logoutBtn" class="btn ghost" onclick="location.href='logout.php'">Sair</button>

      <button id="openChatBtn" class="btn ghost" onclick="location.href='chat.php'">💬 Chat</button>

<span id="chatBadge" class="badge" style="display:none;margin-left:8px">0</span>

    </div>

  </header>

  <main class="main-grid">

    <aside class="sidebar light">

      <h3>Vereadores / Setores</h3>

      <ul id="councilList" class="council-list">

**<?php** foreach ($councilors as $c) echo "<li>" . htmlspecialchars($c) . "</li>"; **?>**

      </ul>

    </aside>

    <section class="content light">

      <div class="controls">

        <button id="tabRegister" class="tab active">Registrar visitante</button>

        <button id="tabPending" class="tab">Aguardando aprovação</button>

        <button id="tabPresent" class="tab">Presentes</button>

        <button id="tabReports" class="tab">Relatórios</button>

      </div>

      <div id="panelRegister" class="panel active">

        <form id="visitorForm" class="visitor-form">

          <div class="autocomplete-wrap">

            <input id="v\_name" name="v\_name" placeholder="Nome do visitante" autocomplete="off" required>

            <div id="v\_suggestions" class="suggestions hidden"></div>

          </div>

          <input id="v\_doc" name="v\_doc" placeholder="Documento (CPF - opcional)" maxlength="14">

          <input id="v\_reason" name="v\_reason" placeholder="Motivo" required>

          <select id="v\_council" name="v\_council" required>

            <option value="">-- Selecione Vereador/Setor --</option>

**<?php** foreach ($councilors as $c) echo "<option value=\"" . htmlspecialchars($c) . "\">" . htmlspecialchars($c) . "</option>"; **?>**

          </select>

          <button type="submit" class="btn primary">Registrar Entrada (Adicionar Visita)</button>

        </form>

        <div class="note muted">Visitas adicionadas aparecem em "Aguardando aprovação".</div>

      </div>

      <div id="panelPending" class="panel">

        <h3>Aguardando aprovação</h3>

        <table class="table" id="pendingTable">

          <thead><tr><th>Nome</th><th>Documento</th><th>Vereador / Setor</th><th>Entrada</th><th>Ações</th></tr></thead>

          <tbody></tbody>

        </table>

      </div>

      <div id="panelPresent" class="panel">

        <h3>Pessoas presentes</h3>

        <table class="table" id="presentTable">

          <thead><tr><th>Nome</th><th>Documento</th><th>Vereador / Setor</th><th>Entrada</th><th>Saída</th><th>Ações</th></tr></thead>

          <tbody></tbody>

        </table>

      </div>

      <div id="panelReports" class="panel">

        <h3>Relatórios</h3>

        <div class="report-controls" style="display:flex;gap:12px;align-items:center;margin-bottom:15px">

          <label>Filtrar por usuário:</label>

          <select id="reportUserFilter"><option value="all">Todos</option></select>

          <label>Período:</label>

          <input type="date" id="reportFrom"> até <input type="date" id="reportTo">

          <button id="generatePdf" class="btn primary">Gerar PDF</button>

          <div class="relatorio-acoes">

            <button id="btn-limpar-relatorios" class="btn-limpar">🧾 Limpar Tudo do Relatório (apagar filtrados)</button>

          </div>

        </div>

        <table class="table" id="reportTable">

          <thead><tr><th>Data</th><th>Visitante</th><th>Documento</th><th>Vereador</th><th>Ação</th><th>Responsável</th></tr></thead>

          <tbody></tbody>

        </table>

      </div>

    </section>

  </main>

  <div id="toast-container" class="toast-container"></div>

</body>

<script>

function updateTopBadge(){

  fetch('chat\_api.php?action=poll\_unread').then(r=>r.json()).then(j=>{

    if (!j.ok) return;

    const b = document.getElementById('chatBadge');

    if (!b) return;

    if (j.unread && j.unread>0) { b.style.display='inline-block'; b.textContent = j.unread; }

    else b.style.display='none';

  });

}

updateTopBadge();

setInterval(updateTopBadge, 5000);

</script>

</html>

// script.js

document.addEventListener('DOMContentLoaded', () => {

  // elementos comuns

  const clockEl = document.getElementById('clock');

  if (clockEl) {

    function updateClock(){ clockEl.textContent = new Date().toLocaleTimeString(); }

    updateClock(); setInterval(updateClock, 1000);

  }

  // abas

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

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

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

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

  function switchTab(panelId, tabId){

    document.querySelectorAll('.panel').forEach(p=>p.classList.remove('active'));

    const panel = document.getElementById(panelId);

    if(panel) panel.classList.add('active');

    document.querySelectorAll('.tab').forEach(t=>t.classList.remove('active'));

    const tab = document.getElementById(tabId);

    if(tab) tab.classList.add('active');

  }

  if(tabRegister) tabRegister.addEventListener('click', ()=> switchTab('panelRegister','tabRegister'));

  if(tabPending) tabPending.addEventListener('click', ()=> switchTab('panelPending','tabPending'));

  if(tabPresent) tabPresent.addEventListener('click', ()=> switchTab('panelPresent','tabPresent'));

  if(tabReports) tabReports.addEventListener('click', ()=> { switchTab('panelReports','tabReports'); loadReportUsers(); renderReport(); });

  // ---------------------- RECEPÇÃO ----------------------

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

  const v\_name = document.getElementById('v\_name');

  const v\_doc = document.getElementById('v\_doc');

  const v\_reason = document.getElementById('v\_reason');

  const v\_council = document.getElementById('v\_council');

  const pendingTable = document.querySelector('#pendingTable tbody');

  const presentTable = document.querySelector('#presentTable tbody');

  if (v\_doc) {

    v\_doc.addEventListener('input', (e)=>{

      let nums = (e.target.value||'').replace(/\D/g,'').slice(0,11);

      // mask

      nums = nums.replace(/(\d{3})(\d)/,'$1.$2').replace(/(\d{3})(\d)/,'$1.$2').replace(/(\d{3})(\d{1,2})$/,'$1-$2');

      e.target.value = nums;

    });

  }

  if (visitorForm) {

    visitorForm.addEventListener('submit', (ev)=>{

      ev.preventDefault();

      const name = (v\_name && v\_name.value || '').trim();

      const doc = (v\_doc && v\_doc.value || '').trim();

      const reason = (v\_reason && v\_reason.value || '').trim();

      const council = (v\_council && v\_council.value || '').trim();

      if (!name || !council || !reason) { showToast('Preencha todos os campos','error'); return; }

      fetch('api.php?action=add\_visitor', {

        method: 'POST',

        headers:{'Content-Type':'application/x-www-form-urlencoded'},

        body: `name=${encodeURIComponent(name)}&doc=${encodeURIComponent(doc)}&council=${encodeURIComponent(council)}&reason=${encodeURIComponent(reason)}`

      }).then(r=>r.json()).then(j=>{

        if (j.ok) {

          showToast('Visitante registrado e aguardando aprovação');

          visitorForm.reset();

          renderLists();

        } else showToast(j.msg || 'Erro','error');

      });

    });

  }

  // actions globales

  window.approveVisitor = function(id){

    fetch('api.php?action=approve', { method:'POST', headers:{'Content-Type':'application/x-www-form-urlencoded'}, body:`id=${id}` })

      .then(r=>r.json()).then(j=>{ if (j.ok) { showToast('Entrada aprovada'); renderLists(); } else showToast(j.msg,'error'); });

  };

// função global para negar (colocar junto das outras window.\* functions)

window.denyVisitor = function(id) {

  if (!confirm('Tem certeza que deseja negar este visitante?')) return;

  fetch('api.php?action=deny', {

    method: 'POST',

    headers: { 'Content-Type': 'application/x-www-form-urlencoded' },

    body: `id=${encodeURIComponent(id)}`

  })

  .then(r => r.json())

  .then(j => {

    if (j.ok) {

      showToast('Visita negada com sucesso');

      // atualiza listas (mesma função que já existe)

      renderLists();

    } else {

      showToast(j.msg || 'Erro ao negar visita', 'error');

    }

  })

  .catch(() => showToast('Erro de conexão ao negar visita', 'error'));

};

  window.registerExit = function(id){

    fetch('api.php?action=exit', { method:'POST', headers:{'Content-Type':'application/x-www-form-urlencoded'}, body:`id=${id}` })

      .then(r=>r.json()).then(j=>{ if (j.ok) { showToast('Saída registrada'); renderLists(); } else showToast(j.msg,'error'); });

  };

  window.removeVisitor = function(id){

    if(!confirm('Remover visitante?')) return;

    // We don't have a specific delete endpoint — but can reuse delete\_filtered with council and id? Simpler: call exit to mark left then remove via API? For now implement as "mark left" to avoid deletion.

    // Alternative: implement delete (but server side doesn't have delete single). So call exit as last resource:

    fetch('api.php?action=exit', { method:'POST', headers:{'Content-Type':'application/x-www-form-urlencoded'}, body:`id=${id}` })

      .then(r=>r.json()).then(j=>{ if (j.ok) { showToast('Visitante marcado como saiu'); renderLists(); } else showToast(j.msg,'error'); });

  };

  // carregar listas

  function renderLists(){

    // pending

    fetch('api.php?action=list\_visitors&filter=pending').then(r=>r.json()).then(j=>{

      if (!j.ok) return;

      if (pendingTable) {

        pendingTable.innerHTML='';

        j.data.forEach(v=>{

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

          tr.innerHTML = `<td>${escapeHtml(v.name)}</td><td>${escapeHtml(v.doc||'')}</td><td>${escapeHtml(v.council)}</td><td>${escapeHtml(v.added\_at)}</td>

            <td>

              <button class="ghost" onclick="approveVisitor(${v.id})">Aprovar</button>

              <button class="ghost" onclick="removeVisitor(${v.id})">Remover</button>

            </td>`;

          pendingTable.appendChild(tr);

        });

      }

    });

    // present

    fetch('api.php?action=list\_visitors&filter=present').then(r=>r.json()).then(j=>{

      if (!j.ok) return;

      if (presentTable) {

        presentTable.innerHTML='';

        j.data.forEach(v=>{

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

          tr.innerHTML = `<td>${escapeHtml(v.name)}</td><td>${escapeHtml(v.doc||'')}</td><td>${escapeHtml(v.council)}</td><td>${escapeHtml(v.entered\_at||'-')}</td><td>${escapeHtml(v.left\_at||'-')}</td>

            <td><button class="ghost" onclick="registerExit(${v.id})">Registrar Saída</button></td>`;

          presentTable.appendChild(tr);

        });

      }

    });

    // report users list

    loadReportUsers();

  }

  // ---------------------- RELATÓRIO ----------------------

  const reportTableBody = document.querySelector('#reportTable tbody');

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

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

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

  function loadReportUsers(){

    if (!reportUserFilter) return;

    fetch('api.php?action=get\_users').then(r=>r.json()).then(j=>{

      if (!j.ok) return;

      reportUserFilter.innerHTML = '<option value="all">Todos</option>';

      j.data.forEach(u => {

        const opt = document.createElement('option');

        opt.value = u.username; opt.textContent = u.fullName;

        reportUserFilter.appendChild(opt);

      });

    });

  }

  function renderReport(){

    const user = (reportUserFilter && reportUserFilter.value) ? reportUserFilter.value : 'all';

    const from = (reportFrom && reportFrom.value) ? reportFrom.value : '';

    const to = (reportTo && reportTo.value) ? reportTo.value : '';

    // call API list\_visitors without filter, but with dates and user

    const qs = `action=list\_visitors&from=${encodeURIComponent(from)}&to=${encodeURIComponent(to)}&user=${encodeURIComponent(user)}`;

    fetch('api.php?' + qs).then(r=>r.json()).then(j=>{

      if (!j.ok) return;

      if (reportTableBody) {

        reportTableBody.innerHTML = '';

        // j.data is list of visitor rows — we need to expand actions (added/entered/left)

        j.data.forEach(rec => {

          // push row for added

          const added = rec.added\_at || rec.added\_at;

          const tbody = reportTableBody;

          // action added

          const tr1 = document.createElement('tr');

          tr1.innerHTML = `<td>${escapeHtml(rec.added\_at)}</td><td>${escapeHtml(rec.name)}</td><td>${escapeHtml(rec.doc||'')}</td><td>${escapeHtml(rec.council)}</td><td>Adicionado</td><td>${escapeHtml(rec.added\_by||'--')}</td>`;

          tbody.appendChild(tr1);

          // entered

          if (rec.entered\_at) {

            const tr2 = document.createElement('tr');

            tr2.innerHTML = `<td>${escapeHtml(rec.entered\_at)}</td><td>${escapeHtml(rec.name)}</td><td>${escapeHtml(rec.doc||'')}</td><td>${escapeHtml(rec.council)}</td><td>Aprovado/Entrou</td><td>${escapeHtml(rec.approved\_by||'--')}</td>`;

            tbody.appendChild(tr2);

          }

          if (rec.left\_at) {

            const tr3 = document.createElement('tr');

            tr3.innerHTML = `<td>${escapeHtml(rec.left\_at)}</td><td>${escapeHtml(rec.name)}</td><td>${escapeHtml(rec.doc||'')}</td><td>${escapeHtml(rec.council)}</td><td>Saiu</td><td>${escapeHtml(rec.exited\_by||'--')}</td>`;

            tbody.appendChild(tr3);

          }

        });

      }

    });

  }

  // limpar relatórios — APAGAR APENAS OS REGISTROS QUE ESTÃO FILTRADOS

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

  if (btnLimpar) {

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

      if (!confirm('Tem certeza que deseja apagar DO BANCO apenas os registros que estão no filtro atual?')) return;

      const user = (reportUserFilter && reportUserFilter.value) ? reportUserFilter.value : 'all';

      const from = (reportFrom && reportFrom.value) ? reportFrom.value : '';

      const to = (reportTo && reportTo.value) ? reportTo.value : '';

      fetch('api.php?action=delete\_filtered', {

        method:'POST',

        headers:{'Content-Type':'application/x-www-form-urlencoded'},

        body: `user=${encodeURIComponent(user)}&from=${encodeURIComponent(from)}&to=${encodeURIComponent(to)}`

      }).then(r=>r.json()).then(j=>{

        if (j.ok) {

          showToast('Registros filtrados apagados: ' + j.deleted);

          renderLists(); renderReport();

        } else showToast(j.msg || 'Erro','error');

      });

    });

  }

  // ---------------------- VEREADOR (polling & UI) ----------------------

if (document.getElementById('vereadorPendingList')) {

  // Funções que chamam a API (assumem api.php?action=...)

  window.approveVisitor = function(id){

    fetch('api.php?action=approve', {

      method:'POST',

      headers:{'Content-Type':'application/x-www-form-urlencoded'},

      body:`id=${encodeURIComponent(id)}`

    }).then(r=>r.json()).then(j=>{

      if (j.ok) {

        showToast('Entrada aprovada', 'success');

        loadVereador(); // atualizar lists

      } else showToast(j.msg || 'Erro ao aprovar','error');

    }).catch(()=> showToast('Erro de comunicação','error'));

  };

  window.denyVisitor = function(id){

    if (!confirm('Tem certeza que deseja negar este visitante?')) return;

    fetch('api.php?action=deny', {

      method:'POST',

      headers:{'Content-Type':'application/x-www-form-urlencoded'},

      body:`id=${encodeURIComponent(id)}`

    }).then(r=>r.json()).then(j=>{

      if (j.ok) {

        showToast('Visita negada com sucesso', 'success');

        loadVereador();

      } else showToast(j.msg || 'Erro ao negar','error');

    }).catch(()=> showToast('Erro de comunicação','error'));

  };

  window.registerExit = function(id){

    fetch('api.php?action=exit', {

      method:'POST',

      headers:{'Content-Type':'application/x-www-form-urlencoded'},

      body:`id=${encodeURIComponent(id)}`

    }).then(r=>r.json()).then(j=>{

      if (j.ok) {

        showToast('Saída registrada', 'success');

        loadVereador();

      } else showToast(j.msg || 'Erro ao registrar saída','error');

    }).catch(()=> showToast('Erro de comunicação','error'));

  };

  // Renderiza a lista pendente e presente como "cards"

  function loadVereador() {

    // pendentes para este vereador

    fetch('api.php?action=my\_pending\_for\_vereador').then(r=>r.json()).then(j=>{

      if (!j.ok) { console.warn('pendentes vazio'); return; }

      const container = document.getElementById('vereadorPendingList');

      container.innerHTML = '';

      if (j.data.length === 0) {

        container.innerHTML = '<div class="small-muted">Nenhuma visita pendente.</div>';

        return;

      }

      j.data.forEach(v => {

        // v should contain: id, name, doc, added\_at, reason

        const row = document.createElement('div');

        row.className = 'notif-row';

        row.setAttribute('data-id', v.id);

        row.innerHTML = `

          <div>

            <h4>${escapeHtml(v.name)}</h4>

            <div class="small-muted">${escapeHtml(v.council || '')}</div>

          </div>

          <div>

            <div class="small-muted">Motivo</div>

            <div>${escapeHtml(v.reason || '-')}</div>

          </div>

          <div>

            <div class="small-muted">Entrada</div>

            <div>${escapeHtml(v.added\_at || '-')}</div>

          </div>

          <div class="actions">

            <button class="btn-accept" onclick="approveVisitor(${v.id})">Aceitar</button>

            <button class="btn-deny" onclick="denyVisitor(${v.id})">Negar</button>

          </div>

        `;

        container.appendChild(row);

      });

    }).catch(e=> console.error(e));

    // presentes atribuídos a este vereador

    fetch('api.php?action=my\_present\_for\_vereador').then(r=>r.json()).then(j=>{

      if (!j.ok) return;

      const container = document.getElementById('vereadorPresentList');

      container.innerHTML = '';

      if (j.data.length === 0) {

        container.innerHTML = '<div class="small-muted">Nenhuma pessoa presente.</div>';

        return;

      }

      j.data.forEach(v => {

        const row = document.createElement('div');

        row.className = 'notif-row';

        row.setAttribute('data-id', v.id);

        row.innerHTML = `

          <div>

            <h4>${escapeHtml(v.name)}</h4>

            <div class="small-muted">${escapeHtml(v.council || '')}</div>

          </div>

          <div>

            <div class="small-muted">Motivo</div>

            <div>${escapeHtml(v.reason || '-')}</div>

          </div>

          <div>

            <div class="small-muted">Entrada</div>

            <div>${escapeHtml(v.entered\_at || '-')}</div>

          </div>

          <div class="actions">

            <button class="btn-accept" onclick="registerExit(${v.id})">Registrar Saída</button>

          </div>

        `;

        container.appendChild(row);

      });

    }).catch(e=> console.error(e));

  }

  // chama logo e depois a cada 5s (polling)

  loadVereador();

  setInterval(loadVereador, 5000);

}

  // ---------------------- util ----------------------

  function showToast(text, type = 'success'){

    const container = document.getElementById('toast-container');

    if(!container) return;

    const t = document.createElement('div');

    t.className = 'toast ' + (type === 'error' ? 'error' : 'success');

    t.innerHTML = `<strong>${text}</strong>`;

    container.appendChild(t);

    setTimeout(()=>{ t.style.opacity = 0; setTimeout(()=>t.remove(), 400); }, 3000);

  }

  function escapeHtml(s){

    if (s === null || s === undefined) return '';

    return String(s).replace(/[&<>"']/g, function(m){ return ({'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;',"'":'&#39;'}[m]); });

  }

  // inicial

  renderLists();

  loadReportUsers();

// Mantém o banco ativo — ping a cada 4 minutos

setInterval(() => {

  fetch('keep\_alive.php')

    .then(r => r.json())

    .then(j => {

      if (j.ok) console.log('Banco ativo:', j.hora);

    })

    .catch(() => console.warn('Ping falhou'));

}, 240000); // 240000 ms = 4 minutos

});

:root{

    --bg:#f8fafc;

    --card:#ffffff;

    --muted:#6b7280;

    --accent:#1e88e5; /\* Azul principal \*/

    --accent-700:#1565c0; /\* Azul escuro \*/

    --glass: rgba(30,136,229,0.06);

    --success:#16a34a;

    --danger:#ef4444;

    /\* Sombra refinada com mais profundidade \*/

    --shadow: 0 10px 30px rgba(30,136,229,0.15), 0 0 8px rgba(30,136,229,0.05);

}

\*{

    box-sizing:border-box;

    font-family:Inter,ui-sans-serif,system-ui,-apple-system,'Segoe UI',Roboto,Arial;

    transition: all 0.3s cubic-bezier(0.2, 0.8, 0.2, 1); /\* Transição global suave \*/

}

body{

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

    margin:0;

    background:var(--bg);

    color:#111;

}

/\* ==================================== \*/

/\* 1. LAYOUT GERAL E EFEITOS DE PROFUNDIDADE \*/

/\* ==================================== \*/

.top-header{

    display:flex;

    align-items:center;

    gap:16px;

    padding:18px 24px;

    background:var(--card);

    box-shadow:var(--shadow);

}

.brasao{

    width:64px;

    height:auto;

    border-radius:8px;

}

.header-title{

    font-weight:700;

}

.login-main{

    display:flex;

    justify-content:center;

    align-items:center;

    height:calc(100vh - 88px);

    padding:24px;

    perspective: 1500px;

}

.login-card{

    display:flex;

    gap:28px;

    max-width:1000px;

    width:100%;

    /\* MUDANÇA CRÍTICA: Efeito de Vidro Fosco (Glassmorphism) \*/

    background: rgba(255, 255, 255, 0.75); /\* Fundo branco um pouco mais opaco \*/

    backdrop-filter: blur(12px);           /\* Mais desfoque \*/

    -webkit-backdrop-filter: blur(12px);

    border-radius:24px; /\* Borda mais suave e moderna \*/

    padding:40px;

    box-shadow: 0 20px 50px rgba(30,136,229,0.3), 0 0 15px rgba(30,136,229,0.1); /\* Sombra mais intensa \*/

    transition: transform 0.8s cubic-bezier(0.2, 0.8, 0.2, 1);

    transform-style: preserve-3d;

    border: 1px solid rgba(255, 255, 255, 0.4); /\* Borda de vidro sutil \*/

}

.login-card:hover {

    transform: scale(1.01); /\* Escala sutil ao passar o mouse \*/

}

/\* ... (Seu código .illustration e .chars existente) ... \*/

.illustration {

  flex:1;

  display:flex;

  align-items:center;

  justify-content:center;

  background: var(--accent-light);

  border-radius:18px; /\* Mais arredondado \*/

  padding:24px;

  position: relative;

  overflow: hidden;

  box-shadow: inset 0 0 25px rgba(30, 136, 229, 0.5); /\* Brilho interno mais forte \*/

}

.chars {

  width:320px;

  height:220px;

  position:relative;

  /\* Animação de fundo pulsante \*/

  background: repeating-linear-gradient(

    45deg,

    rgba(255,255,255,0.08),

    rgba(255,255,255,0.08) 12px,

    transparent 12px,

    transparent 24px

  );

  animation: scan 20s linear infinite;

}

@keyframes scan {

  from { background-position: 0 0; }

  to { background-position: 480px 480px; }

}

.char{

    position:absolute;

    transition:all .5s ease;

    border-radius:12px;

}

/\* ... (Cores e posições dos chars) ... \*/

.char1{width:90px;height:90px;background:#7c5cff;left:40px;top:20px;transform:rotate(-8deg)}

.char2{width:110px;height:110px;background:#ff6b6b;left:10px;bottom:0;border-radius:55px 55px 20px 20px}

.char3{width:80px;height:160px;background:#ffd166;right:40px;bottom:20px;border-radius:30px}

.char4{width:60px;height:100px;background:#0a0a0a;left:140px;bottom:20px;border-radius:6px}

/\* ESTILOS DE LOGIN INTERATIVO (PEEP) \*/

.chars.peep .char1{

  transform: translateY(10px) rotate(-4deg);

}

.chars.peep .char2{

  transform: translateX(-10px) translateY(0);

}

.chars.peep .char3{

  transform: translateX(5px) translateY(-5px);

}

.chars.peep .char4{

  transform: translateX(-5px) rotate(5deg);

}

.chars.hide .char1{transform:translateY(40px) scale(.6);opacity:.3}

.chars.hide .char2{transform:translateY(60px) scale(.6);opacity:.3}

.chars.hide .char3{transform:translateY(60px) scale(.6);opacity:.3}

.chars.hide .char4{transform:translateY(40px) scale(.5);opacity:.2}

/\* Foco e entrada \*/

.login-box{flex:1;padding:8px}

.muted{color:var(--muted)}

.login-box label{display:block;margin-top:12px;font-size:13px;color:var(--muted)}

.login-box input{

    width:100%;

    padding:14px; /\* Mais preenchimento \*/

    margin-top:6px;

    border-radius:12px; /\* Mais arredondado \*/

    border:2px solid #e6edf3; /\* Borda mais grossa \*/

    background:var (--card);

    transition: all 0.3s cubic-bezier(0.2, 0.8, 0.2, 1);

}

.login-box input:focus {

    outline: none;

    border-color: var(--accent); /\* Cor de foco mais suave \*/

    box-shadow: 0 0 15px rgba(30,136,229,0.3); /\* Brilho maior no foco \*/

}

.hint{margin-top:8px}

/\* ==================================== \*/

/\* 2. BOTÕES (Aprimoramento do Acabamento) \*/

/\* ==================================== \*/

.btn{

    display:inline-flex; /\* Ajuste para centralizar ícones se usar \*/

    align-items: center;

    justify-content: center;

    padding:10px 18px;

    border-radius:10px;

    border:none;

    cursor:pointer;

    font-weight:700; /\* Mais espessura \*/

    font-size: 15px;

    transition: all 0.25s ease;

}

/\* Botão Primário (o mais Foda!) \*/

.btn.primary{

    background:linear-gradient(90deg,var(--accent),var(--accent-700));

    color:white;

    /\* Sombra com mais destaque e brilho interno sutil \*/

    box-shadow: 0 8px 20px rgba(30,136,229,0.3), inset 0 0 10px rgba(255, 255, 255, 0.3);

    /\* Animação de pulso sutil para o botão mais importante \*/

    animation: pulse 2s infinite cubic-bezier(0.66, 0.01, 0.34, 0.99);

}

.btn.primary:hover {

    transform: translateY(-2px) scale(1.01);

    box-shadow: 0 12px 30px rgba(30,136,229,0.4), inset 0 0 15px rgba(255, 255, 255, 0.5);

}

@keyframes pulse {

  0%, 100% { box-shadow: 0 8px 20px rgba(30,136,229,0.3), inset 0 0 10px rgba(255, 255, 255, 0.3); }

  50% { box-shadow: 0 8px 25px rgba(30,136,229,0.5), inset 0 0 10px rgba(255, 255, 255, 0.5); }

}

.btn.ghost{

    background:transparent;

    border:1px solid #e6edf3;

    color:var(--muted);

    padding:10px 14px;

    border-radius:10px;

}

.btn.ghost:hover {

    border-color: var(--accent);

    color: var(--accent-700);

    box-shadow: 0 2px 8px rgba(30,136,229,0.1);

}

/\* Botão GIGANTE de Adicionar \*/

.visitor-form .btn.primary {

  grid-column: 1 / -1; /\* Ocupa a linha inteira \*/

  font-size: 1.2em;

  padding: 16px;

  margin-top: 8px; /\* Espaçamento extra \*/

}

/\* ==================================== \*/

/\* 3. TOPBAR E INFORMAÇÕES DE STATUS \*/

/\* ==================================== \*/

.topbar{

    display:flex;

    justify-content:space-between;

    align-items:center;

    padding:12px 24px;

    background:transparent;

    position: sticky;

    top: 0;

    z-index: 1000;

}

.topbar.light{

    background:var(--card);

    box-shadow:var(--shadow);

}

.brand{

    display:flex;

    align-items:center;

    gap:16px; /\* Mais espaço \*/

}

.title{

    font-weight:900; /\* Mais peso \*/

    font-size: 1.3em;

}

.subtitle{

    font-size:14px; /\* Maior \*/

    color:var(--muted);

}

.top-actions{

    display:flex;

    align-items:center;

    gap:16px;

}

/\* Estilo de Glassmorphism para os elementos de status \*/

.clock{

    padding:10px 14px;

    border-radius:10px;

    background: rgba(30,136,229,0.1); /\* Um pouco mais escuro \*/

    backdrop-filter: blur(5px);

    font-weight:800;

    color: var(--accent-700);

    border: 1px solid rgba(30,136,229,0.2);

}

.user-display{

    padding:10px 14px;

    border-radius:10px;

    background:linear-gradient(90deg,#eef7ff,#f3f9ff);

    font-weight:800;

    color:var(--accent-700);

    border: 1px solid #d0e7ff;

}

/\* ==================================== \*/

/\* 4. CONTEÚDO PRINCIPAL (MAIN GRID) \*/

/\* ==================================== \*/

.main-grid{

    display:grid;

    grid-template-columns:280px 1fr; /\* Sidebar um pouco mais larga \*/

    gap:24px; /\* Mais espaçamento \*/

    padding:24px;

}

/\* NOVO: Estilo de Contêiner com Borda de Neon Sutil \*/

.sidebar, .content {

    background: var(--card);

    padding: 20px; /\* Mais preenchimento \*/

    border-radius: 18px; /\* Mais arredondado \*/

    box-shadow: var(--shadow);

    /\* Borda de acento sutil, criando a sensação de "tecnológico" \*/

    border: 1px solid rgba(30, 136, 229, 0.1);

    transition: all 0.3s ease-in-out;

}

/\* NOVO: Efeito de Brilho ao Passar o Mouse nos Contêineres \*/

.sidebar:hover, .content:hover {

    box-shadow: 0 16px 40px rgba(30,136,229,0.3), 0 0 15px rgba(30,136,229,0.15);

    transform: translateY(-2px);

    border-color: rgba(30, 136, 229, 0.3);

}

/\* ==================================== \*/

/\* 5. SIDEBAR (Lista de Vereadores) \*/

/\* ==================================== \*/

.toast-container {

  position: fixed;

  bottom: 20px;

  right: 20px;

  z-index: 9999;

}

.toast {

  background: #333;

  color: #fff;

  padding: 10px 14px;

  border-radius: 6px;

  margin-top: 8px;

  opacity: 0.9;

  font-size: 14px;

  transition: all 0.3s ease;

}

.toast.success { background: #28a745; }

.toast.error { background: #dc3545; }

button.ghost.danger {

  background-color: #dc3545;

  color: white;

}

button.ghost.danger:hover {

  background-color: #b02a37;

}

.council-list{

    list-style: none; /\* Remove marcadores \*/

    padding: 0;

    max-height:70vh;

    overflow-y:auto;

    margin-top:12px;

}

/\* Estilo de lista Foda \*/

.council-list li{

    padding: 12px;

    border-radius: 10px;

    margin-bottom: 8px;

    background: linear-gradient(90deg,#fbfdff,#f6f9ff);

    border: 1px solid #eef3f9;

    color: #0b2740;

    font-weight: 600;

    transition: all 0.3s ease;

    cursor: pointer;

    text-shadow: 0 1px 1px rgba(255, 255, 255, 0.5); /\* Sombra sutil no texto \*/

}

.council-list li:hover {

    background: linear-gradient(90deg,#e6f4ff,#dff0ff);

    box-shadow: 0 4px 12px rgba(30,136,229,0.15);

    transform: translateX(4px);

}

/\* Estado Ativo/Selecionado (Simulação de um LED aceso) \*/

.council-list li.selected {

    background: linear-gradient(90deg, var(--accent), var(--accent-700));

    color: white;

    font-weight: 700;

    box-shadow: 0 0 15px rgba(30,136,229,0.5);

    border-color: var(--accent);

    transform: translateX(0);

    text-shadow: none;

}

/\* ==================================== \*/

/\* 6. CONTEÚDO (TABS e Formulário) \*/

/\* ==================================== \*/

.controls{

    display:flex;

    gap:12px; /\* Mais espaço entre as abas \*/

    margin-bottom:20px;

    border-bottom: 2px solid #eef3f9; /\* Linha de separação \*/

    padding-bottom: 8px;

}

.tab{

    background:#fbfdff;

    padding:10px 16px;

    border-radius:10px 10px 0 0; /\* Borda arredondada no topo \*/

    border:1px solid #eef3f9;

    border-bottom: none;

    cursor:pointer;

    font-weight: 600;

    color: var(--muted);

    transition: all 0.2s ease;

}

.tab:hover:not(.active) {

    background: #e6f4ff;

}

.tab.active{

    background:linear-gradient(90deg,var(--accent),var(--accent-700));

    color:white;

    transform: translateY(2px); /\* Efeito de "subida" \*/

    box-shadow: 0 4px 15px rgba(30,136,229,0.3);

    border-color: var(--accent);

}

.panel{display:none}

.panel.active{

    display:block;

    /\* Animação para a entrada do painel \*/

    animation: floatIn 0.5s cubic-bezier(0.2, 0.8, 0.2, 1);

}

@keyframes floatIn {

  from { opacity: 0; transform: translateY(15px); }

  to { opacity: 1; transform: translateY(0); }

}

.visitor-form{

    display:grid;

    grid-template-columns:1fr 1fr; /\* Layout em 2 colunas \*/

    gap:15px; /\* Mais espaçamento \*/

    align-items:center;

}

.visitor-form input,.visitor-form select{

    padding:12px;

    border-radius:10px;

    border:2px solid #e6edf3;

    background:transparent;

}

.visitor-form input:focus, .visitor-form select:focus {

    border-color: var(--accent);

    box-shadow: 0 0 8px rgba(30,136,229,0.2);

    outline: none;

}

.note{margin-top:10px}

/\* ==================================== \*/

/\* 7. TABELAS (Melhorando a Leitura) \*/

/\* ==================================== \*/

.table{

    width:100%;

    border-collapse:separate; /\* Para bordas arredondadas \*/

    border-spacing: 0;

    margin-top:15px;

    border-radius: 12px;

    overflow: hidden; /\* Garante que as bordas arredondadas funcionem \*/

}

.table th,.table td{

    padding:14px; /\* Mais preenchimento \*/

    border-bottom:1px solid #eef3f9;

    text-align:left;

}

.table thead th{

    background: var(--glass); /\* Glassmorphism no cabeçalho \*/

    color: #0b2740; /\* Cor mais escura que o muted \*/

    font-weight: 700;

    text-transform: uppercase;

    font-size: 0.85em;

    letter-spacing: 0.5px;

}

.table tbody tr:last-child td {

    border-bottom: none;

}

.table tbody tr:hover {

    background-color: #f7f9fc; /\* Sutil destaque ao passar o mouse na linha \*/

}

.footer{

    padding:18px;

    text-align:center;

    margin-top:24px;

    border-top:1px solid #eef3f9;

    color:var(--muted);

}

/\* ... (Seu código para toasts, relatorio-acoes e autocomplete) ... \*/

.toast-container{position:fixed;right:20px;top:20px;z-index:9999}

.toast{background:rgba(255,255,255,0.98);padding:12px 16px;margin-bottom:10px;border-radius:10px;box-shadow:0 8px 30px rgba(11,18,28,0.12);display:flex;gap:12px;align-items:center;border-left:4px solid var(--accent-700)}

.toast.success{border-left-color:var(--success)}

.toast.error{border-left-color:var(--danger)}

.relatorio-acoes {

  margin-top: 0px;

  display: flex;

  flex-wrap: wrap;

  gap: 10px;

  justify-content: center;

}

.btn-limpar {

  background: linear-gradient(90deg, #e53935, #c62828);

  color: #fff;

  border: none;

  padding: 10px 18px;

  border-radius: 10px;

  cursor: pointer;

  font-weight: 600;

  transition: 0.3s;

}

.btn-limpar:hover {

  transform: translateY(-2px) scale(1.01);

  box-shadow: 0 4px 15px rgba(229, 57, 53, 0.4);

}

/\* 🔍 Autocomplete \*/

.autocomplete-wrap {

  position: relative;

  grid-column: 1 / 2; /\* Garante que o wrap siga o layout \*/

}

.suggestions {

  position: absolute;

  left: 0;

  right: 0;

  top: calc(100% + 5px); /\* Abaixo do input com um pequeno espaço \*/

  background: var(--card);

  border: 1px solid #e6edf3;

  box-shadow: 0 10px 30px rgba(11,18,28,0.1);

  z-index: 99;

  border-radius: 10px;

  max-height: 250px;

  overflow-y: auto;

}

.suggestions.hidden {

  display: none;

}

.suggestion-item {

  padding: 12px;

  cursor: pointer;

  transition: background 0.2s;

}

.suggestion-item:hover {

  background: #e8f2ff;

}

@media (max-width:1000px){

    .main-grid{grid-template-columns:1fr;padding:12px}

    .visitor-form{grid-template-columns:1fr;gap:12px}

    .visitor-form > \* { grid-column: 1 / -1 !important; } /\* Força todos a ficarem em uma coluna no mobile \*/

    .login-card{flex-direction:column}

}

/\* --- centraliza e limita largura do layout principal --- \*/

.main-grid {

  max-width: 1200px;

  margin: 0 auto;        /\* centraliza horizontalmente \*/

  grid-template-columns: 320px 1fr; /\* fixar largura da sidebar para evitar overflow \*/

}

/\* força a sidebar a ficar compacta e o content a ajustar ao conteúdo \*/

.sidebar {

  width: 320px;

  min-width: 240px;

}

/\* o content não precisa ocupar 100vh quando houver pouco conteúdo \*/

.content {

  min-height: 120px; /\* ajusta conforme necessário \*/

}

/\* botão vermelho (negativo) consistente com tema \*/

.btn-deny, button.ghost.danger {

  background: linear-gradient(45deg,#e74c3c,#b02a37);

  color: #fff;

  border-radius: 8px;

  padding: 6px 10px;

  border: none;

  font-weight: 700;

}

/\* botão ghost/danger hover \*/

button.ghost.danger:hover, .btn-deny:hover {

  transform: translateY(-2px);

  box-shadow: 0 6px 18px rgba(176,42,55,0.25);

}

/\* evita que colunas da tabela quebrem o layout \*/

.table td, .table th {

  word-break: break-word;

  vertical-align: middle;

}

/\* deixa o card lateral com largura mínima condizente ao conteúdo \*/

.sidebar.light {

  width: auto;

  max-width: 360px;

}

/\* ================== MELHOR VISUAL - PAINEL DO VEREADOR ================== \*/

/\* Centraliza o card de notificações e ajusta espaçamento \*/

.main-grid {

  display: flex;

  justify-content: center;

  align-items: flex-start;

  padding: 40px 20px;

}

/\* Card principal \*/

.content.light {

  background: #ffffff;

  width: 100%;

  max-width: 850px;

  padding: 40px;

  border-radius: 20px;

  box-shadow: 0 10px 35px rgba(0, 40, 100, 0.08);

  transition: all 0.3s ease;

}

/\* Efeito leve ao passar o mouse (deixa moderno) \*/

.content.light:hover {

  transform: scale(1.01);

  box-shadow: 0 12px 40px rgba(0, 40, 100, 0.12);

}

/\* Títulos \*/

.content.light h3 {

  font-size: 1.6rem;

  color: #0b3d91;

  font-weight: 700;

  margin-bottom: 12px;

}

.content.light h4 {

  font-size: 1.1rem;

  color: #333;

  margin-top: 30px;

  margin-bottom: 10px;

}

/\* Texto informativo \*/

.content.light .note {

  color: #777;

  font-size: 0.95rem;

  margin-bottom: 20px;

}

/\* Tabelas \*/

.table {

  width: 100%;

  border-collapse: collapse;

  border-radius: 10px;

  overflow: hidden;

  background: #f8fbff;

}

.table th {

  background: #e6f0ff;

  padding: 10px;

  color: #002a5e;

  font-weight: 600;

  text-transform: uppercase;

  font-size: 13px;

}

.table td {

  padding: 12px;

  border-bottom: 1px solid #e2e8f0;

  color: #333;

}

/\* Botões \*/

button.ghost {

  background: linear-gradient(180deg, #0078ff, #0057c2);

  color: #fff;

  border: none;

  border-radius: 8px;

  padding: 6px 12px;

  cursor: pointer;

  font-weight: 600;

  transition: 0.3s;

}

button.ghost:hover {

  transform: translateY(-2px);

  opacity: 0.95;

}

/\* Botão de negar \*/

button.ghost.danger {

  background: linear-gradient(180deg, #d9534f, #b52b27);

}

/\* Efeito hover mais bonito \*/

button.ghost.danger:hover {

  background: linear-gradient(180deg, #b52b27, #d9534f);

}

/\* Responsivo \*/

@media (max-width: 768px) {

  .content.light {

    padding: 24px;

  }

  .table th, .table td {

    font-size: 12px;

    padding: 8px;

  }

}

/\* ======= Estilos específicos para painel vereador (cards notificação) ======= \*/

.vereador-container { max-width: 900px; margin: 18px auto; padding: 18px; }

.card-notif { background: var(--card); border-radius: 12px; padding: 16px; box-shadow: var(--shadow); margin-bottom: 16px; }

.notif-row { display: grid; grid-template-columns: 1fr 220px 160px 180px; gap: 12px; align-items: center; padding: 12px; border-radius: 10px; border: 1px solid #eef3f9; background: linear-gradient(90deg,#fff,#fbfdff); margin-bottom: 10px; }

.notif-row h4 { margin: 0; font-size: 16px; }

.notif-meta { color: var(--muted); font-size: 13px; }

.actions { display:flex; gap:8px; justify-content:flex-end; }

.btn-accept { background: linear-gradient(90deg,var(--accent),var(--accent-700)); color:#fff; border:0; padding:8px 12px; border-radius:8px; cursor:pointer; font-weight:700; }

.btn-deny { background: linear-gradient(45deg,#ff6b6b,#c92b2b); color:#fff; border:0; padding:8px 12px; border-radius:8px; cursor:pointer; font-weight:700; }

.small-muted { font-size:13px; color:var(--muted); }

@media (max-width:900px){ .notif-row { grid-template-columns: 1fr; gap:8px; } .actions { justify-content:flex-start; } }

**<?php**

// vereador.php

require\_once 'config.php';

if (session\_status() === PHP\_SESSION\_NONE) session\_start();

if (empty($\_SESSION['user'])) {

    header('Location: index.php');

    exit;

}

$user = $\_SESSION['user'];

if ($user['role'] !== 'vereador' && $user['role'] !== 'admin') {

    echo "Acesso negado.";

    exit;

}

**?>**

<!doctype html>

<html lang="pt-BR">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <title>Painel - Vereador</title>

  <link rel="stylesheet" href="style.css">

  <!-- Pequeno CSS local para ajustar a área de notificações (pode também ir ao style.css) -->

  <style>

    /\* Painel centralizado e card leve para notificações \*/

    .vereador-container {

      max-width: 900px;

      margin: 18px auto;

      padding: 18px;

    }

    .card-notif {

      background: var(--card);

      border-radius: 12px;

      padding: 16px;

      box-shadow: var(--shadow);

      margin-bottom: 16px;

    }

    .notif-row {

      display: grid;

      grid-template-columns: 1fr 220px 160px 180px; /\* nome / motivo / entrada / ações \*/

      gap: 12px;

      align-items: center;

      padding: 12px;

      border-radius: 10px;

      border: 1px solid #eef3f9;

      background: linear-gradient(90deg, #fff, #fbfdff);

      margin-bottom: 10px;

    }

    .notif-row h4 { margin: 0; font-size: 16px; }

    .notif-meta { color: var(--muted); font-size: 13px; }

    .actions { display:flex; gap:8px; justify-content:flex-end; }

    .btn-accept { background: linear-gradient(90deg,var(--accent),var(--accent-700)); color:#fff; border:0; padding:8px 12px; border-radius:8px; cursor:pointer; font-weight:700; }

    .btn-deny { background: linear-gradient(45deg,#ff6b6b,#c92b2b); color:#fff; border:0; padding:8px 12px; border-radius:8px; cursor:pointer; font-weight:700; }

    .small-muted { font-size:13px; color:var(--muted); }

    /\* responsivo \*/

    @media (max-width:900px){

      .notif-row { grid-template-columns: 1fr; gap:8px; }

      .actions { justify-content:flex-start; }

    }

  </style>

  <script defer src="script.js"></script>

</head>

<body class="panel-page">

  <header class="topbar light">

    <div class="brand">

      <img src="https://tse3.mm.bing.net/th/id/OIP.NeJbj2QckKlAAfZ0YlkgUgHaJw?cb=12&pid=Api" alt="Brasão" class="brasao">

      <div>

        <div class="title">Painel do Vereador</div>

        <div class="subtitle">Câmara Municipal</div>

      </div>

    </div>

    <div class="top-actions">

      <div id="clock" class="clock">--:--:--</div>

      <div id="userDisplay" class="user-display">**<?php** echo htmlspecialchars($user['fullName']); **?>**</div>

      <button class="btn ghost" onclick="location.href='chat.php'">💬 Chat</button>

<span id="chatNotif" class="chat-notif hidden">●</span>

      <button id="logoutBtn" class="btn ghost" onclick="location.href='logout.php'">Sair</button>

    </div>

  </header>

  <main class="main-grid">

    <section class="content light vereador-container">

      <div class="card-notif">

        <h3>Notificações</h3>

        <div class="note muted">Aqui aparecem visitantes atribuídos a você. Você pode aceitar (autorizar entrada) ou negar.</div>

      </div>

      <div class="card-notif" id="pendingArea">

        <h4>Visitantes pendentes para você</h4>

        <div id="vereadorPendingList" style="margin-top:12px;"></div>

      </div>

      <div class="card-notif" id="presentArea">

        <h4>Seus presentes</h4>

        <div id="vereadorPresentList" style="margin-top:12px;"></div>

      </div>

    </section>

  </main>

  <div id="toast-container" class="toast-container"></div>

</body>

<script>

function updateTopBadge(){

  fetch('chat\_api.php?action=poll\_unread').then(r=>r.json()).then(j=>{

    if (!j.ok) return;

    const b = document.getElementById('chatBadge');

    if (!b) return;

    if (j.unread && j.unread>0) { b.style.display='inline-block'; b.textContent = j.unread; }

    else b.style.display='none';

  });

}

updateTopBadge();

setInterval(updateTopBadge, 5000);

</script>

</html>