// app.js - lógica para cargar menu.json y renderizar la carta

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

const menuUrl = 'menu.json';

// fetch menu.json

let menuData;

try {

const res = await fetch(menuUrl);

if(!res.ok) throw new Error('No se pudo cargar menu.json');

menuData = await res.json();

} catch (err) {

console.error(err);

alert('Error al cargar menu.json. Asegúrate de servir los archivos desde un servidor (no funciona con file:// en algunos navegadores).');

return;

}

const originalMenu = JSON.parse(JSON.stringify(menuData)); // deep copy

// Helpers

const el = sel => document.querySelector(sel);

const els = sel => Array.from(document.querySelectorAll(sel));

function getAllAllergens(data){

const set = new Set();

data.sections.forEach(sec => sec.items.forEach(it => it.allergens.forEach(a => set.add(a))));

return Array.from(set).filter(Boolean).sort();

}

function renderAllergenFilters(){

const container = el('#allergenFilters'); container.innerHTML = '';

const allergens = getAllAllergens(menuData);

if(allergens.length===0){ container.innerHTML = '<span class="small">Sin alérgenos en el listado</span>'; return; }

allergens.forEach(a =>{

const id = 'alg\_'+a.replace(/[^a-z0-9]/gi,'');

const wrap = document.createElement('label'); wrap.className='filter-chip';

wrap.innerHTML = `<input type="checkbox" data-allergen="${a}" id="${id}" /> <span class="small">${a}</span>`;

container.appendChild(wrap);

wrap.querySelector('input').addEventListener('change', applyFilters);

})

}

function renderMenu(){

const list = el('#menuList'); list.innerHTML='';

const q = el('#search').value.trim().toLowerCase();

const activeAllergens = els('#allergenFilters input[type=checkbox]:checked').map(i=>i.dataset.allergen);

let visibleCount = 0;

menuData.sections.forEach(section=>{

// section title

const secTitle = document.createElement('h2'); secTitle.textContent = section.name; secTitle.style.margin='18px 0 8px';

list.appendChild(secTitle);

section.items.forEach(item=>{

const text = (item.name + ' ' + item.ingredients.join(' ')).toLowerCase();

if(q && !text.includes(q)) return;

if(activeAllergens.length && item.allergens.some(a=> activeAllergens.includes(a))) return;

visibleCount++;

const card = document.createElement('article'); card.className='dish';

card.innerHTML = `

<div class="meta">

<h3>${escapeHtml(item.name)}</h3>

<div class="small">${section.name} • <span class="price">${item.price.toFixed(2)}€</span></div>

<div class="tags" data-id="${item.id}"></div>

</div>

<div style="text-align:right; min-width:160px">

<div class="small">Alérgenos:</div>

<div class="tags" style="justify-content:flex-end">${item.allergens.map(a=>`<span class="tag allergen">${escapeHtml(a)}</span>`).join(' ')}</div>

<div style="margin-top:10px; display:flex; gap:8px; justify-content:flex-end">

<button class="btn ghost editBtn" data-id="${item.id}">Editar</button>

</div>

</div>

`;

// render ingredients tags

const tagsDiv = card.querySelector('.tags[data-id]');

item.ingredients.forEach(ing =>{

const s = document.createElement('span'); s.className='tag'; s.textContent = ing; tagsDiv.appendChild(s);

})

list.appendChild(card);

})

})

el('#visibleCount').textContent = visibleCount;

// attach edit handlers

els('.editBtn').forEach(b=> b.addEventListener('click', onEdit));

}

function applyFilters(){ renderMenu(); }

function onEdit(e){

const id = e.currentTarget.dataset.id;

let found=null;

menuData.sections.forEach(sec=> sec.items.forEach(it=> { if(it.id===id) found=it; }));

if(!found) return alert('Plato no encontrado');

const editor = document.createElement('div'); editor.className='edit-area';

editor.innerHTML = `

<label class="small">Ingredientes (separados por coma)</label>

<input type="text" id="ingInput" value="${escapeHtml(found.ingredients.join(', '))}" />

<label class="small" style="margin-top:8px">Alérgenos (separados por coma)</label>

<input type="text" id="algInput" value="${escapeHtml(found.allergens.join(', '))}" />

<div style="margin-top:8px; display:flex; gap:8px; justify-content:flex-end">

<button class="btn ghost" id="cancelEdit">Cancelar</button>

<button class="btn primary" id="saveEdit">Guardar</button>

</div>

`;

const card = e.currentTarget.closest('.dish');

els('.edit-area').forEach(a=>a.remove());

card.appendChild(editor);

editor.querySelector('#cancelEdit').addEventListener('click', ()=>{ editor.remove(); });

editor.querySelector('#saveEdit').addEventListener('click', ()=>{

const newIngs = editor.querySelector('#ingInput').value.split(',').map(s=>s.trim()).filter(Boolean);

const newAlgs = editor.querySelector('#algInput').value.split(',').map(s=>s.trim()).filter(Boolean);

found.ingredients = newIngs;

found.allergens = newAlgs;

editor.remove();

renderAllergenFilters();

renderMenu();

});

}

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

function downloadJSON(){

const blob = new Blob([JSON.stringify(menuData, null, 2)], {type: 'application/json'});

const url = URL.createObjectURL(blob);

const a = document.createElement('a'); a.href=url; a.download='menu\_actualizado.json'; document.body.appendChild(a); a.click(); a.remove(); URL.revokeObjectURL(url);

}

function openSample(){

const w = window.open('','Carta JSON','width=700,height=600,scrollbars=yes');

w.document.write('<pre style="white-space:pre-wrap;font-family:monospace;padding:12px">'+escapeHtml(JSON.stringify(menuData, null, 2))+'</pre>');

w.document.title = 'menu.json (vista)';

}

function resetData(){ if(confirm('Restablecer la carta a la versión original? Se perderán cambios no exportados.')){ menuData = JSON.parse(JSON.stringify(originalMenu)); renderAllergenFilters(); renderMenu(); } }

function downloadSplitFiles(){

// descarga los tres archivos: index.html, menu.json, app.js (basado en el estado actual)

// Para facilitar la descarga, se creará un archivo por cada contenido conocido

fetch('index.html').then(r=>r.text()).then(indexContent=>{

downloadTextAsFile(indexContent, 'index.html');

downloadTextAsFile(JSON.stringify(menuData, null, 2), 'menu.json');

fetch('app.js').then(r2=>r2.text()).then(appContent=>{

downloadTextAsFile(appContent, 'app.js');

});

}).catch(err => {

// si falla la petición (por ejemplo file://), intentamos generar básicos

downloadTextAsFile('<index.html contenido no disponible>', 'index.html');

downloadTextAsFile(JSON.stringify(menuData, null, 2), 'menu.json');

downloadTextAsFile('// app.js contenido no disponible', 'app.js');

});

}

function downloadTextAsFile(text, filename){ const blob = new Blob([text],{type:'text/plain'}); const url=URL.createObjectURL(blob); const a=document.createElement('a'); a.href=url; a.download=filename; document.body.appendChild(a); a.click(); a.remove(); URL.revokeObjectURL(url); }

// events

el('#search').addEventListener('input', renderMenu);

el('#downloadBtn').addEventListener('click', downloadJSON);

el('#resetBtn').addEventListener('click', resetData);

el('#openSampleBtn').addEventListener('click', openSample);

el('#splitFilesBtn').addEventListener('click', downloadSplitFiles);

// initial render

renderAllergenFilters();

renderMenu();

});