<!doctype html>

<div class="small">Tip: this demo uses static/sample data. To pull real feeds use a server proxy or RSS->JSON service (CORS is restricted on many endpoints).</div>

</main>

<footer>Built for brevity — edit the <code>posts</code> array inside the HTML to add real items.</footer>

</div>

<script>

// ======= SAMPLE POSTS =======

const samplePosts = [

{source: 'Twitter', user: '@alice', time: '2h', text: 'Launched a tiny JS project — lightweight aggregator!'},

{source: 'Instagram', user: '@bobby', time: '1d', text: 'Sunset walk 🌇 #evening'},

{source: 'LinkedIn', user: 'Cara Doe', time: '3d', text: 'Sharing a quick note about product-market fit.'}

];

// Renders posts into the grid

function renderPosts(posts){

const out = document.getElementById('feed');

out.innerHTML = '';

posts.forEach(p => {

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

el.className = 'card';

el.innerHTML = `

<div class="meta"><span class="source">${escapeHtml(p.source)}</span><strong>${escapeHtml(p.user)}</strong><span style="margin-left:auto">${escapeHtml(p.time)}</span></div>

<div class="text">${escapeHtml(p.text)}</div>

`;

out.appendChild(el);

});

}

// Basic HTML escape

function escapeHtml(s){

if(!s && s !== 0) return '';

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

}

// Try to fetch JSON from URL if user provided one. Expect an array of {source,user,time,text}

async function loadFromUrl(url){

try{

const res = await fetch(url);

if(!res.ok) throw new Error('Fetch failed');

const j = await res.json();

if(Array.isArray(j)) return j;

// common wrapper keys

if(Array.isArray(j.items)) return j.items;

if(Array.isArray(j.posts)) return j.posts;

throw new Error('JSON not an array of posts');

}catch(err){

alert('Could not load JSON: '+err.message+'\nUsing sample data.');

return samplePosts;

}

}

document.getElementById('load').addEventListener('click', async ()=>{

const url = document.getElementById('jsonInput').value.trim();

if(!url){ renderPosts(samplePosts); return; }

const posts = await loadFromUrl(url);

renderPosts(posts);

});

// initial render

renderPosts(samplePosts);

</script>

</body>

</html>