

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8" />
```

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

```
<title>Single-file Blogging Platform</title>
```

```
<style>
```

```
:root{--bg:#0f1724;--card:#0b1220;--accent:#06b6d4;--muted:#94a3b8;--glass:rgba(255,255,255,0.03)}
```

```
*{box-sizing:border-box}
```

```
html,body{height:100%;margin:0;font-family:Inter,ui-sans-serif,system-ui,Segoe UI,Roboto,'Helvetica Neue',Arial;color:#e6eef6;background:linear-gradient(180deg,#071426 0%,#082032 100%);}
```

```
.app{max-width:1100px;margin:28px auto;padding:20px}
```

```
header{display:flex;gap:16px;align-items:center;justify-content:space-between}
```

```
h1{font-size:20px;margin:0}
```

```
.controls{display:flex;gap:8px;align-items:center}
```

```
input[type=search],input[type=text]{padding:8px 12px;border-radius:10px;border:1px solid rgba(255,255,255,0.06);background:rgba(255,255,255,0.03);color:inherit}
```

```
button{background:var(--accent);border:none;padding:8px 12px;border-radius:10px;color:#032;cursor:pointer;font-weight:600}
```

```
main{display:grid;grid-template-columns:360px 1fr;gap:18px;margin-top:18px}/* Sidebar / composer */
```

```
.sidebar{background:var(--card);border-radius:12px;padding:14px;box-shadow:0 6px 18px rgba(2,6,23,0.6)}
```

```
label{display:block;font-size:12px;color:var(--muted);margin-bottom:6px}
```

```
.field{margin-bottom:12px}
```

```
.title-input{width:100%;padding:10px;border-radius:8px;border:1px solid rgba(255,255,255,0.04);background:transparent;color:inherit}
```

```
.tags-input{display:flex;gap:8px;flex-wrap:wrap}
```

```
.tag{background:var(--glass);padding:6px 8px;border-radius:8px;font-size:12px;color:var(--muted)}

textarea{width:100%;min-height:160px;padding:10px;border-radius:8px;border:1px solid
rgba(255,255,255,0.04);background:transparent;color:inherit;resize:vertical}

.small{font-size:12px;color:var(--muted)}

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


/* Post list */

.posts{background:linear-gradient(180deg, rgba(255,255,255,0.02), transparent);border-
radius:12px;padding:14px;min-height:360px}

.post-row{display:flex;flex-direction:column;padding:12px;border-radius:10px;margin-
bottom:10px;background:rgba(255,255,255,0.015)}

.post-head{display:flex;align-items:center;gap:10px;justify-content:space-between}

.post-meta{font-size:12px;color:var(--muted)}

.post-title{font-weight:700;margin:6px 0}

.post-body{color:#dbeaf6;font-size:14px;white-space:pre-wrap}

.row-actions{display:flex;gap:8px}

.ghost{background:transparent;border:1px solid rgba(255,255,255,0.04);padding:6px 10px;border-
radius:8px;color:var(--muted);cursor:pointer}


/* Modal */

.modal{position:fixed;inset:0;display:none;align-items:center;justify-content:center;background:linear-
gradient(0deg, rgba(2,6,23,0.6), rgba(2,6,23,0.6));}

.modal.open{display:flex}

.card{background:linear-gradient(180deg,#071428, #071a2a);padding:18px;border-radius:12px;max-
width:900px;width:94%;}


/* responsive */

@media (max-width:900px){main{grid-template-columns:1fr;}

.controls{flex-wrap:wrap} }
```

```
/* tiny helpers */

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

.flex{display:flex;gap:8px;align-items:center}

.pill{padding:6px 10px;border-radius:999px;background:rgba(255,255,255,0.02);font-size:12px}


</style>

</head>

<body>

  <div class="app">

    <header>

      <div>

        <h1>Single-file Blogging Platform</h1>

        <div class="muted" style="font-size:13px;margin-top:6px">Create, edit, delete & save posts
        locally (localStorage). Export/Import JSON.</div>

        </div>

        <div class="controls">

          <input id="search" type="search" placeholder="Search posts..." />

          <button id="btn-export">Export JSON</button>

          <button id="btn-import" title="Import posts from JSON">Import JSON</button>

          <input id="file-input" type="file" accept="application/json" style="display:none" />

        </div>

      </header><main>

<aside class="sidebar">

  <div class="field">

    <label>Title</label>

    <input id="title" class="title-input" placeholder="Post title" />

  </div>

  <div class="field">
```

```
<label>Tags (comma separated)</label>

<input id="tags" class="title-input" placeholder="e.g. javascript,web" />

</div>

<div class="field">

  <label>Content (markdown-lite)</label>

  <textarea id="content" placeholder="Write your post here. Use blank lines for paragraphs.">
</textarea>

  <div class="small">Tip: use &lt;h1> lines starting with # for headings, blank lines become
paragraphs.</div>

</div>

<div class="actions">

  <button id="btn-save">Save Post</button>

  <button class="ghost" id="btn-clear">Clear</button>

</div>

</aside>


<section>

  <div class="posts" id="post-list">

    <!-- posts injected here -->

  </div>

</section>

</main>


</div> <!-- Modal for preview / edit --> <div id="modal" class="modal" aria-hidden="true">

  <div class="card">

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

      <strong id="modal-title">Preview</strong>
```

```
<div class="flex"><button class="ghost" id="modal-edit">Edit</button><button id="modal-close"
class="ghost">Close</button></div>
```

```
</div>
```

```
<hr style="margin:12px 0;border:none;border-top:1px solid rgba(255,255,255,0.03)"/>
```

```
<div id="modal-body"></div>
```

```
</div>
```

```
</div> <script>
```

```
// Simple single-file blogging platform
```

```
() => {
```

```
  const LS_KEY = 'sf_blog_posts_v1';
```

```
  const titleEl = document.getElementById('title');
```

```
  const tagsEl = document.getElementById('tags');
```

```
  const contentEl = document.getElementById('content');
```

```
  const btnSave = document.getElementById('btn-save');
```

```
  const btnClear = document.getElementById('btn-clear');
```

```
  const postList = document.getElementById('post-list');
```

```
  const searchEl = document.getElementById('search');
```

```
  const btnExport = document.getElementById('btn-export');
```

```
  const btnImport = document.getElementById('btn-import');
```

```
  const fileInput = document.getElementById('file-input');
```

```
  const modal = document.getElementById('modal');
```

```
  const modalBody = document.getElementById('modal-body');
```

```
  const modalTitle = document.getElementById('modal-title');
```

```
  const modalClose = document.getElementById('modal-close');
```

```
  const modalEdit = document.getElementById('modal-edit');
```

```
  let posts = load();
```

```
  let editingId = null;
```

```
function uid(){return 'p_' + Math.random().toString(36).slice(2,9)}
```

```
function now(){return new Date().toISOString()}
```

```
function saveToStorage(){localStorage.setItem(LS_KEY, JSON.stringify(posts))}
```

```
function load(){try{const raw=localStorage.getItem(LS_KEY);return raw?  
JSON.parse(raw):defaultPosts()}catch(e){console.error(e);return []}}
```

```
function defaultPosts(){return [{id:uid(),title:'Welcome to your blog',tags:  
['welcome','intro'],content:'This is a sample post. Edit or delete it.\n\n- Use the form to the left to create  
new posts.\n- Posts are saved to your browser (localStorage).',created:now(),updated:now()}]}
```

```
function renderList(filter=""){  
  
  postList.innerHTML = "";  
  
  const q = filter.trim().toLowerCase();  
  
  const filtered = posts.filter(p=>{  
  
    if(!q) return true;  
  
    return p.title.toLowerCase().includes(q) || p.content.toLowerCase().includes(q) || (p.tags||  
[]).join(' ').toLowerCase().includes(q);  
  
  });  
  
  if(filtered.length===0){postList.innerHTML = '<div class="muted" style="padding:18px">No posts  
yet. Create one using the form.</div>';return}  
  
  filtered.sort((a,b)=>new Date(b.updated)-new Date(a.updated));  
  
  for(const p of filtered){  
  
    const el = document.createElement('div'); el.className='post-row';  
  
    const head = document.createElement('div'); head.className='post-head';  
  
    const left = document.createElement('div');  
  
    const title = document.createElement('div'); title.className='post-title'; title.textContent = p.title  
|| '(untitled)';  
  
    const meta = document.createElement('div'); meta.className='post-meta'; meta.textContent =  
`${(p.tags||[]).join(', ')} • ${new Date(p.updated).toLocaleString()}`;
```

```

    left.appendChild(title); left.appendChild(meta);

    const actions = document.createElement('div'); actions.className='row-actions';

    const btnView = document.createElement('button'); btnView.className='ghost';
    btnView.textContent='Preview'; btnView.onclick = ()=>openModal(p);

    const btnEdit = document.createElement('button'); btnEdit.className='ghost';
    btnEdit.textContent='Edit'; btnEdit.onclick = ()=>startEdit(p);

    const btnDel = document.createElement('button'); btnDel.className='ghost';
    btnDel.textContent='Delete'; btnDel.onclick = ()=>deletePost(p.id);

    actions.appendChild(btnView); actions.appendChild(btnEdit); actions.appendChild(btnDel);

    head.appendChild(left); head.appendChild(actions);

    const body = document.createElement('div'); body.className='post-body'; body.innerHTML =
    renderMarkdownPreview(p.content).slice(0,600) + (p.content.length>600? '...':" ");

    el.appendChild(head); el.appendChild(body);

    postList.appendChild(el);
  }
}

function renderMarkdownPreview(text){
  if(!text) return "";

  // very small markdown-like renderer: headings (#), lists ( - ), paragraphs

  const lines = text.split('\n');

  const out = [];

  let inList = false;

  for(let raw of lines){
    const line = raw.trim();

    if(line.startsWith('#')){ const lvl = Math.min(6, line.match(/^#+/)[0].length);
    out.push(`<h${lvl}>${escapeHtml(line.slice(lvl).trim())}</h${lvl}>`); continue}

```

```

    if(line.startsWith('- ')){ if(!inList){ inList=true; out.push('<ul>') } out.push('<li>' +
escapeHtml(line.slice(2)) + '</li>'); continue }

    if(inList){ inList=false; out.push('</ul>') }

    if(line==="){ out.push('<p>&nbsp;</p>'); continue }

    out.push('<p>'+escapeHtml(line)+'</p>')

}

if(inList) out.push('</ul>');

return out.join('\n')

}

```

```

function escapeHtml(s){return s.replaceAll('&','&amp;').replaceAll('<','&lt;').replaceAll('>','&gt;')}

```

```

function clearForm()
{titleEl.value="";tagsEl.value="";contentEl.value="";editingId=null;btnSave.textContent='Save Post'}

```

```

function startEdit(p){editingId = p.id; titleEl.value = p.title; tagsEl.value = (p.tags||[]).join(',');
contentEl.value = p.content; window.scrollTo({top:0,behavior:'smooth'}); btnSave.textContent='Update
Post'}

```

```

function deletePost(id){ if(!confirm('Delete this post?')) return; posts = posts.filter(p=>p.id!==id);
saveToStorage(); renderList(searchEl.value); }

```

```

function openModal(p){ modalTitle.textContent = p.title || '(untitled)'; modalBody.innerHTML =
renderMarkdownPreview(p.content) + `<div class="muted" style="margin-top:16px">Tags: ${p.tags||
[]}.join(', ')} • Updated: ${new Date(p.updated).toLocaleString()}</div>`; modal.classList.add('open');
modal.setAttribute('aria-hidden','false'); modalEdit.onclick = ()=>{modal.classList.remove('open');
startEdit(p)} }

```

```

function upsertPost(){

const t = titleEl.value.trim();

const c = contentEl.value.trim();

```



```

const tags = tagsEl.value.split(',').map(s=>s.trim()).filter(Boolean);

if(!t && !c){alert('Please add a title or content.');
```

```

return}

if(editingId){

    const idx = posts.findIndex(p=>p.id===editingId); if(idx===-1) return; posts[idx].title=t;
posts[idx].content=c; posts[idx].tags=tags; posts[idx].updated=now();

    }else{

        posts.push({id:uid(),title:t,content:c,tags:tags,created:now(),updated:now()});

    }

    saveToStorage(); clearForm(); renderList(searchEl.value);

}

// export/import

btnExport.onclick = ()=>{

    const data = JSON.stringify(posts, null, 2);

    const blob = new Blob([data],{type:'application/json'});

    const url = URL.createObjectURL(blob);

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

}

btnImport.onclick = ()=>fileInput.click();

fileInput.onChange = async (e)=>{

    const f = e.target.files && e.target.files[0]; if(!f) return; try{ const txt = await f.text(); const
imported = JSON.parse(txt); if(!Array.isArray(imported)) throw new Error('Invalid format');

    // merge imported (avoid id collisions)

    const existingIds = new Set(posts.map(p=>p.id));

    for(const p of imported){ if(!p.id || existingIds.has(p.id)) p.id = uid(); posts.push(p); }

    saveToStorage(); renderList(""); alert('Imported ' + imported.length + ' posts');

}catch(err){ alert('Import failed: ' + err.message) }

```

```

}

// modal controls

modalClose.onclick = ()=>{ modal.classList.remove('open'); modal.setAttribute('aria-hidden','true')
}

modal.addEventListener('click', (e)=>{ if(e.target===modal) modalClose.click() })

// events

btnSave.onclick = upsertPost;

btnClear.onclick = clearForm;

searchEl.oninput = ()=>renderList(searchEl.value);

// initial render

renderList("");

// small extras for keyboard shortcuts

window.addEventListener('keydown', (e)=>{

  if((e.ctrlKey||e.metaKey) && e.key.toLowerCase()=== 's'){ e.preventDefault(); upsertPost(); }

}))

// expose for debugging (optional)

window.sfBlog = {posts,saveToStorage,renderList}

})();

</script></body>

</html>

```