

EXPERIMENT 3.2

AIM:

Library Management UI with Search, Add, and Remove Book Functionality

CODE:

```
<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8" />

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

  <title>Library Management — HTML</title>

  <style>

    :root{

      --bg:#f6f7fb; --card:#ffffff; --muted:#6b7280; --accent:#0f172a;

      --danger:#ef4444; --success:#10b981; --border:#e6e9ef;

      font-family: Inter, ui-sans-serif, system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica

Neue", Arial;

      font-size:14px;

    }

    *{box-sizing:border-box}

    body{margin:0;background:linear-gradient(180deg,var(--bg),#ffffff);color:var(--

accent);padding:20px}

    .container{max-width:1100px;margin:0 auto}
```

```
header{display:flex;gap:16px;align-items:end;justify-content:space-between;margin-
bottom:18px}

h1{margin:0;font-size:24px}

p.lead{margin:0;color:var(--muted);font-size:13px}

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

input[type="search"], input[type="text"], input[type="number"], select{

padding:8px 10px;border:1px solid var(--border);border-radius:10px;background:var(--
card);

outline:none;font-size:13px;min-width:180px;

}

input:focus, select:focus{box-shadow:0 0 3px rgba(15,23,42,0.06)}

form.card{display:grid;gap:8px;padding:12px;border-radius:12px;background:var(--
card);border:1px solid var(--border);}

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

.form-row > *{flex:1}

button{cursor:pointer;border-radius:10px;padding:8px 12px;border:0;background:var(--
accent);color:white;font-weight:600}

button.ghost{background:white;color:var(--accent);border:1px solid var(--
border);padding:7px 10px}

button.danger{background:var(--danger)}

.table-wrap{overflow:auto;margin-top:12px;border-radius:10px;border:1px solid var(--
border);background:var(--card)}

table{width:100%;border-collapse:collapse;font-size:13px}

thead th{background:transparent;padding:10px 12px;text-align:left;color:var(--muted);font-
weight:600;vertical-align:middle}
```

```
tbody td{padding:10px 12px;border-top:1px solid var(--border);vertical-align:middle}

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

.small{font-size:12px}

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

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

.pill{padding:6px 8px;border-radius:999px;border:1px solid var(--border);background:white}

.sort-ind{font-size:11px;color:var(--muted);margin-left:6px}

.error{color:var(--danger);font-size:12px;margin-top:4px}

@media (max-width:720px){

  .form-row{flex-direction:column}

  header{flex-direction:column;align-items:flex-start;gap:8px}

  .controls{width:100%;flex-wrap:wrap}

}

</style>

</head>

<body>

<div class="container">

  <header>

    <div>

      <h1> Library Management</h1>

      <p class="lead">Search, add and remove books — stored in your browser

(localStorage).</p>

    </div>

    <div class="controls" aria-hidden="false">
```

```

        <input id="search" type="search" placeholder="Search title, author, ISBN or category..."
    />

    <label class="muted small">Page size

        <select id="pagesize" class="pill">

            <option>5</option><option
selected>6</option><option>10</option><option>15</option>

        </select>

    </label>

    <button id="reset" class="ghost" title="Reset sample data">Reset</button>

</div>

</header>

<section>

    <form id="addForm" class="card" autocomplete="off" onsubmit="return false;">

        <div class="form-row">

            <div>

                <label class="small muted">Title *</label>

                <input id="title" type="text" placeholder="e.g., Clean Code" required />

                <div id="err-title" class="error" style="display:none"></div>

            </div>

            <div>

                <label class="small muted">Author *</label>

                <input id="author" type="text" placeholder="e.g., Robert C. Martin" required />

                <div id="err-author" class="error" style="display:none"></div>

            </div>

```

</div>

<div class="form-row">

<div>

<label class="small muted">Year</label>

<input id="year" type="number" min="0" placeholder="e.g., 2008" />

<div id="err-year" class="error" style="display:none"></div>

</div>

<div>

<label class="small muted">ISBN</label>

<input id="isbn" type="text" placeholder="e.g., 9780132350884" />

<div id="err-isbn" class="error" style="display:none"></div>

</div>

<div>

<label class="small muted">Category</label>

<input id="category" type="text" placeholder="e.g., Software" />

</div>

</div>

<div style="display:flex;justify-content:flex-end;margin-top:6px">

<button id="addBtn">+ Add Book</button>

</div>

</form>

</section>

```
<section style="margin-top:12px">
```

```
<div class="table-wrap">
```

```
<table id="booksTable" aria-live="polite">
```

```
<thead>
```

```
<tr>
```

```
<th style="width:30%;cursor:pointer" data-col="title">Title <span class="sort-ind" id="s-title">⬆⬇</span></th>
```

```
<th style="width:25%;cursor:pointer" data-col="author">Author <span class="sort-ind" id="s-author">⬆⬇</span></th>
```

```
<th style="width:8%;cursor:pointer" data-col="year">Year <span class="sort-ind" id="s-year">⬆⬇</span></th>
```

```
<th style="width:15%;">ISBN</th>
```

```
<th style="width:12%;">Category</th>
```

```
<th style="width:10%;text-align:right; padding-right:16px" data-col="addedAt">Added <span class="sort-ind" id="s-addedAt">⬆⬇</span></th>
```

```
</tr>
```

```
</thead>
```

```
<tbody id="tbody">
```

```
<!-- rows injected here -->
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
<div style="display:flex;justify-content:space-between;align-items:center;margin-top:10px">
```

```
<div class="muted small" id="info">Showing 0 of 0 books</div>
```

```

<div class="pager">

  <button id="prev" class="ghost">Prev</button>

  <div class="muted small" id="pageInfo">Page 1 / 1</div>

  <button id="next" class="ghost">Next</button>

</div>

</div>

</section>

</div>

<script>

(function(){

  // --- storage & seed ---

  const KEY = 'library.books.v1';

  const seed = [

    { id: uid(), title: "Clean Code", author: "Robert C. Martin", year:2008,
isbn:"9780132350884", category:"Software", addedAt: Date.now()-432000000 },

    { id: uid(), title: "The Pragmatic Programmer", author: "Andrew Hunt, David Thomas",
year:1999, isbn:"9780201616224", category:"Software", addedAt: Date.now()-259200000 },

    { id: uid(), title: "Atomic Habits", author: "James Clear", year:2018, isbn:"9780735211292",
category:"Self-help", addedAt: Date.now()-86400000 },

  ];

  function uid(){ return

'id_'+(Date.now().toString(36))+'_'+Math.random().toString(36).slice(2,9); }

```

```
// --- state ---

let books = load() || seed.slice();

let query = '';

let sortBy = 'addedAt'; // title, author, year, addedAt

let sortDir = 'desc'; // asc | desc

let page = 1;

let pageSize = Number(document.getElementById('pagesize').value) || 6;


// --- elements ---

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

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

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

const prevBtn = document.getElementById('prev');

const nextBtn = document.getElementById('next');

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

const pageSizeEl = document.getElementById('pagesize');

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

const resetBtn = document.getElementById('reset');


// header sort indicators

const sortIndicators = {

  title: document.getElementById('s-title'),

  author: document.getElementById('s-author'),

  year: document.getElementById('s-year'),

  addedAt: document.getElementById('s-addedAt')}
```



```

};

// initial render

render();

// --- persistence helpers ---

function load(){

  try{ const raw = localStorage.getItem(KEY); return raw ? JSON.parse(raw) : null; }

catch(e){ return null; }

}

function save(){

  try{ localStorage.setItem(KEY, JSON.stringify(books)); } catch(e){}

}

// --- filtering, sorting, pagination ---

function filtered(){

  const q = query.trim().toLowerCase();

  const base = q ? books.filter(b => {

    return (b.title||'').toLowerCase().includes(q)

    || (b.author||'').toLowerCase().includes(q)

    || (b.isbn||'').toLowerCase().includes(q)

    || (b.category||'').toLowerCase().includes(q);

  }) : books.slice();

  base.sort((a,b)=>{

```

```

let av = a[sortBy], bv = b[sortBy];

if (sortBy==='title' || sortBy==='author'){

    av = (av||'').toString().toLowerCase();

    bv = (bv||'').toString().toLowerCase();

} else {

    av = av ?? 0; bv = bv ?? 0;

}

if (av < bv) return sortDir==='asc' ? -1 : 1;

if (av > bv) return sortDir==='asc' ? 1 : -1;

return 0;

});

return base;

}

function render(){

    // update pageSize from select - safe sync

    pageSize = Number(pagesizeEl.value) || 6;


    const list = filtered();

    const total = list.length;

    const totalPages = Math.max(1, Math.ceil(total / pageSize));

    if (page > totalPages) page = totalPages;


    const start = (page-1)*pageSize;

    const pageItems = list.slice(start, start + pageSize);

```

```

// table body

tbody.innerHTML = '';

if (pageItems.length === 0){

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

    tr.innerHTML = '<td colspan="6" style="text-align:center;padding:18px;color:var(--
muted)">No books found.</td>';

    tbody.appendChild(tr);

} else {

    for (const b of pageItems){

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

        const addedRelative = timeAgo(b.addedAt);

        tr.innerHTML = `

        <td title="\${escapeHtml(b.title)}" style="font-weight:600">\${escapeHtml(b.title)}</td>

        <td title="\${escapeHtml(b.author)}">\${escapeHtml(b.author)}</td>

        <td>\${b.year ?? '—'}</td>

        <td>\${b.isbn ? escapeHtml(b.isbn) : '—'}</td>

        <td>\${b.category ? escapeHtml(b.category) : '—'}</td>

        <td style="text-align:right">

            <div style="display:flex;gap:8px;align-items:center;justify-content:flex-end">

                <span class="muted small" title="\${new
Date(b.addedAt).toLocaleString()}">\${addedRelative}</span>

                <button class="ghost" data-remove="\${b.id}" title="Remove">Remove</button>

            </div>

```

```

        </td>

    `;

    tbody.appendChild(tr);

}

}

// info + pager

info.textContent = `Showing ${pageItems.length} of ${total} book${total===1 ? '' : 's'}`;

pageInfo.textContent = `Page ${page} / ${totalPages}`;

prevBtn.disabled = page <= 1;

nextBtn.disabled = page >= totalPages;

// attach remove handlers

tbody.querySelectorAll('button[data-remove]').forEach(btn=>{

    btn.addEventListener('click', (ev)=>{

        const id = btn.getAttribute('data-remove');

        if (!id) return;

        const b = books.find(x => x.id === id);

        const ok = confirm(`Remove "${b ? b.title : 'this book'}"?`);

        if (ok){

            books = books.filter(x => x.id !== id);

            save();

            render();

        }
    });
});

```

```

});

});

// update sort indicators

Object.keys(sortIndicators).forEach(k=>{

  const el = sortIndicators[k];

  if (!el) return;

  if (k === sortBy) el.textContent = sortDir === 'asc' ? '▲' : '▼';

  else el.textContent = '↑↓';

});

}

// --- events ---

document.querySelectorAll('th[data-col]').forEach(th=>{

  th.style.userSelect = 'none';

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

    const col = th.getAttribute('data-col');

    if (!col) return;

    if (sortBy === col){

      sortDir = (sortDir === 'asc') ? 'desc' : 'asc';

    } else {

      sortBy = col;

      sortDir = (col === 'addedAt') ? 'desc' : 'asc';

    }

    page = 1;

```

```
render();

});

});
```

```
searchEl.addEventListener('input', (e)=>{

    query = e.target.value || "";

    page = 1;

    render();

});
```

```
pagesizeEl.addEventListener('change', ()=>{

    page = 1;

    render();

});
```

```
prevBtn.addEventListener('click', ()=>{ if (page>1) { page--; render(); } });

nextBtn.addEventListener('click', ()=>{ page++; render(); });
```

```
// Add book
```

```
addBtn.addEventListener('click', ()=>{

    clearErrors();

    const title = document.getElementById('title').value.trim();

    const author = document.getElementById('author').value.trim();

    const yearRaw = document.getElementById('year').value.trim();

    const isbn = document.getElementById('isbn').value.trim();
```

```
const category = document.getElementById('category').value.trim();

// validation

const errors = {};

if (!title) errors.title = 'Title is required';

if (!author) errors.author = 'Author is required';

let year;

if (yearRaw){

    year = Number(yearRaw);

    const cur = new Date().getFullYear() + 1;

    if (!Number.isInteger(year) || year < 0 || year > cur) errors.year = 'Enter a valid year';

}

if (isbn){

    const dup = books.some(b => b.isbn && b.isbn.trim() === isbn.trim());

    if (dup) errors.isbn = 'ISBN already exists';

}

if (Object.keys(errors).length){

    showErrors(errors);

    return;

}

const newBook = {

    id: uid(),

    title, author,
```

```

    year: year ?? undefined,

    isbn: isbn || undefined,

    category: category || undefined,

    addedAt: Date.now()

  };

  books.unshift(newBook); // newest first

  save();

  // clear form

  document.getElementById('title').value = '';

  document.getElementById('author').value = '';

  document.getElementById('year').value = '';

  document.getElementById('isbn').value = '';

  document.getElementById('category').value = '';

  page = 1;

  render();

});

// Reset sample

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

  if (!confirm('Reset library to sample books? This will overwrite your saved list.')) return;

  books = seed.slice();

  save();

  page = 1;

  render();

});

```


// helpers: errors display

function showErrors(map){

if (map.title) { showEl('err-title', map.title); }

if (map.author) { showEl('err-author', map.author); }

if (map.year) { showEl('err-year', map.year); }

if (map.isbn) { showEl('err-isbn', map.isbn); }

}

function showEl(id, txt){

const el = document.getElementById(id);

if (!el) return;

el.style.display = 'block'; el.textContent = txt;

}

function clearErrors(){

['err-title','err-author','err-year','err-isbn'].forEach(id=>{

const e = document.getElementById(id);

if (e){ e.style.display='none'; e.textContent=''; }

});

}

// utility: relative time in days/hours/minutes

function timeAgo(ts){

const diff = Date.now() - ts;

const mins = Math.round(diff / 60000);

if (mins < 1) return 'just now';

```

    if (mins < 60) return mins + 'm';

    const hours = Math.round(mins / 60);

    if (hours < 24) return hours + 'h';

    const days = Math.round(hours / 24);

    if (days < 30) return days + 'd';

    const months = Math.round(days / 30);

    if (months < 12) return months + 'mo';

    const years = Math.round(months / 12);

    return years + 'y';

}

// escape html

function escapeHtml(s){

    if (!s) return '';

    return s.replace(/[\&<>"]'/g, function(m){ return

{ '&':'&amp;','<':'&lt;','>':'&gt;','\"':'&quot;','\'' : '&#39;'}[m]; });

}

// simple demo: update storage at start

save();

render();

})0;

</script>

</body>

</html>

```


OUTPUT:

AI

NEW

HTML

RUN

 **Library Management**

Search, add and remove books — stored in your browser (localStorage).

Search title, author, ISBN

Page size 6

Reset

Title *

e.g., Clean Code

Author *

e.g., Robert C. Martin

Year

e.g., 2008

ISBN

e.g., 9780132350884

Category

e.g., Software

+ Add Book

Title ↑	Author ↑	Year ↑	ISBN	Category	Added ▼
Atomic Habits	James Clear	2018	9780735211292	Self-help	1d Remove
The Pragmatic Programmer	Andrew Hunt, David Thomas	1999	9780201616224	Software	3d Remove