Name :Nikhil Sharma

Class: 23BCC-(A)

UID : 23BCC70030  
  
  
 FULL STACK

EXPERIMENT 3.2

**Aim**

Build a simple Library Management UI that lets users search books by title/author, add new books, and remove existing books—using only HTML, CSS, and JavaScript (no frameworks).

CODE:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Library Management UI</title>

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

<style>

:root { font-family: system-ui, Arial, sans-serif; }

body { margin: 0; background:#f6f7fb; color:#121212; }

.container { max-width: 900px; margin: 32px auto; padding: 0 16px; }

.card { background:#fff; border-radius:16px; box-shadow: 0 6px 24px rgba(0,0,0,.08); padding: 20px; }

h1 { margin: 0 0 16px; font-size: 24px; }

.grid { display: grid; gap: 16px; grid-template-columns: 1fr; }

.toolbar { display:flex; gap:12px; flex-wrap: wrap; }

input, button, select {

font: inherit; padding: 10px 12px; border-radius: 10px; border: 1px solid #e3e6ef; outline: none;

}

input:focus { border-color:#5b8def; box-shadow:0 0 0 3px rgba(91,141,239,.15); }

button { border: none; cursor: pointer; background:#5b8def; color:#fff; transition:.15s; }

button:hover { filter: brightness(0.95); }

button.secondary { background:#eef2ff; color:#334; }

table { width:100%; border-collapse: collapse; overflow:hidden; border-radius: 12px; }

thead th { text-align:left; background:#f3f5fb; padding:12px; font-weight:600; font-size:14px; }

tbody td { padding:12px; border-top:1px solid #eef1f6; font-size:14px; }

.muted { color:#667085; font-size: 12px; }

.right { text-align:right; }

.form { display:grid; grid-template-columns: 2fr 2fr 1fr auto; gap:10px; }

@media (max-width: 720px) {

.form { grid-template-columns: 1fr; }

.right { text-align:left; }

}

.badge { background:#ecfdf5; color:#065f46; padding:2px 8px; border-radius: 999px; font-size: 12px; }

.empty { text-align:center; padding:24px; color:#667085; }

</style>

</head>

<body>

<div class="container">

<div class="card grid">

<h1>Library Management</h1>

<!-- Search + Reset -->

<div class="toolbar">

<input id="searchInput" type="text" placeholder="Search by title or author..." />

<button class="secondary" id="resetBtn" title="Clear search & form">Reset</button>

<span class="muted" id="countLabel"></span>

</div>

<!-- Add Book Form -->

<form id="addForm" class="form">

<input id="titleInput" type="text" placeholder="Book title" required />

<input id="authorInput" type="text" placeholder="Author" required />

<input id="yearInput" type="number" min="0" max="3000" placeholder="Year" required />

<button type="submit">Add Book</button>

</form>

<!-- Books Table -->

<div class="card" style="padding:0;">

<table>

<thead>

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

<th class="right">Action</th>

</tr>

</thead>

<tbody id="booksTbody"></tbody>

</table>

<div id="emptyState" class="empty" style="display:none;">No books found.</div>

</div>

<span class="muted">Tip: Use search to filter. Add new books with the form. Remove with the “Delete” button.</span>

</div>

</div>

<script>

// --- In-memory data store ---

let books = [

{ id: 1, title: "Clean Code", author: "Robert C. Martin", year: 2008 },

{ id: 2, title: "The Pragmatic Programmer", author: "Andrew Hunt", year: 1999 },

{ id: 3, title: "Introduction to Algorithms", author: "Cormen", year: 2009 },

];

const booksTbody = document.getElementById("booksTbody");

const emptyState = document.getElementById("emptyState");

const searchInput = document.getElementById("searchInput");

const titleInput = document.getElementById("titleInput");

const authorInput = document.getElementById("authorInput");

const yearInput = document.getElementById("yearInput");

const addForm = document.getElementById("addForm");

const resetBtn = document.getElementById("resetBtn");

const countLabel = document.getElementById("countLabel");

// Render books into table

function render(list) {

booksTbody.innerHTML = "";

if (!list.length) {

emptyState.style.display = "block";

updateCount(0, books.length);

return;

}

emptyState.style.display = "none";

list.forEach(b => {

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

const tdTitle = document.createElement("td");

tdTitle.textContent = b.title;

const tdAuthor = document.createElement("td");

tdAuthor.innerHTML = `${b.author} ${isClassic(b.year) ? '<span class="badge">classic</span>' : ''}`;

const tdYear = document.createElement("td");

tdYear.textContent = b.year;

const tdAction = document.createElement("td");

tdAction.className = "right";

const delBtn = document.createElement("button");

delBtn.textContent = "Delete";

delBtn.addEventListener("click", () => removeBook(b.id));

tdAction.appendChild(delBtn);

tr.appendChild(tdTitle);

tr.appendChild(tdAuthor);

tr.appendChild(tdYear);

tr.appendChild(tdAction);

booksTbody.appendChild(tr);

});

updateCount(list.length, books.length);

}

function isClassic(year) {

const y = Number(year);

if (!y) return false;

const now = new Date().getFullYear();

return now - y >= 20; // mark as "classic" if 20+ years old

}

// Add a new book

addForm.addEventListener("submit", (e) => {

e.preventDefault();

const title = titleInput.value.trim();

const author = authorInput.value.trim();

const year = Number(yearInput.value);

if (!title || !author || !year) {

alert("Please fill all fields.");

return;

}

// Optional: prevent exact duplicates (title+author+year)

const dup = books.some(b =>

b.title.toLowerCase() === title.toLowerCase() &&

b.author.toLowerCase() === author.toLowerCase() &&

Number(b.year) === year

);

if (dup) {

alert("This book already exists.");

return;

}

books.push({ id: Date.now(), title, author, year });

addForm.reset();

applySearch(); // re-render with current search filter

});

// Remove a book by id

function removeBook(id) {

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

if (!book) return;

const ok = confirm(`Delete "${book.title}" by ${book.author}?`);

if (!ok) return;

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

applySearch();

}

// Search filtering

searchInput.addEventListener("input", applySearch);

function applySearch() {

const q = searchInput.value.trim().toLowerCase();

if (!q) {

render(books);

return;

}

const filtered = books.filter(b =>

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

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

);

render(filtered);

}

// Reset search + form

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

searchInput.value = "";

addForm.reset();

render(books);

titleInput.focus();

});

function updateCount(showing, total) {

countLabel.textContent = showing

? `Showing ${showing} of ${total} book(s)`

: total

? `No results. (${total} total)`

: `No books yet`;

}

// Initial render

render(books);

</script>

</body>

</html>

OUTPUT:

