<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Advanced Library Management System</title>

<style>

/\* BODY & BACKGROUND \*/

body {

margin: 0;

font-family: Arial, sans-serif;

background: linear-gradient(135deg, #6a11cb, #2575fc);

min-height: 100vh;

display: flex;

justify-content: center;

align-items: center;

overflow-x: hidden;

}

/\* COMMON PAGE STYLING \*/

.page {

background: #fff;

padding: 40px 30px;

border-radius: 20px;

box-shadow: 0 15px 50px rgba(0,0,0,0.2);

width: 400px;

max-width: 90%;

position: relative;

}

/\* FLOATING LABELS \*/

.input-group {

position: relative;

margin-bottom: 25px;

}

.input-group input {

width: 100%;

padding: 15px;

border: 1px solid #ccc;

border-radius: 50px;

outline: none;

font-size: 16px;

background: transparent;

transition: 0.3s;

}

.input-group label {

position: absolute;

left: 20px;

top: 50%;

transform: translateY(-50%);

color: #aaa;

pointer-events: none;

transition: 0.3s;

}

.input-group input:focus + label,

.input-group input:not(:placeholder-shown) + label {

top: -10px;

font-size: 12px;

color: #2575fc;

background: #fff;

padding: 0 5px;

}

.input-group input:focus {

border-color: #2575fc;

box-shadow: 0 0 8px rgba(37,117,252,0.5);

}

/\* BUTTONS \*/

button {

width: 100%;

padding: 15px;

border-radius: 50px;

border: none;

background: #2575fc;

color: white;

font-size: 16px;

font-weight: bold;

cursor: pointer;

transition: 0.3s;

}

button:hover {

background: #6a11cb;

box-shadow: 0 5px 20px rgba(106,17,203,0.4);

}

/\* LINKS \*/

.page p {

margin-top: 20px;

font-size: 14px;

color: #555;

text-align: center;

}

.page p span {

color: #2575fc;

cursor: pointer;

}

.page p span:hover {

text-decoration: underline;

}

/\* DASHBOARD STYLING \*/

#dashboard .dashboard-header {

display:flex;

justify-content:space-between;

align-items:center;

margin-bottom:20px;

}

.tabs {

display:flex;

margin-bottom:15px;

}

.tab-btn {

padding:10px 20px;

border:none;

background:#eee;

cursor:pointer;

margin-right:5px;

border-radius:5px 5px 0 0;

}

.tab-btn.active {background:#2575fc; color:#fff;}

.tab-content {display:block;}

table {width:100%; border-collapse:collapse; margin-top:10px;}

th, td {padding:10px; border:1px solid #ccc; text-align:left;}

th {background:#2575fc; color:#fff; cursor:pointer;}

.modal {display:none; position:fixed; top:0; left:0; width:100%; height:100%; background:rgba(0,0,0,0.5); z-index:100;}

.modal-content {background:#fff; margin:10% auto; padding:20px; width:300px; border-radius:10px; position:relative;}

.close {position:absolute; top:10px; right:15px; font-size:20px; cursor:pointer;}

.search-add {display:flex; justify-content:space-between; margin-bottom:10px;}

.notification {position:fixed; top:10px; right:10px; background:#ff4d4d; color:#fff; padding:10px 20px; border-radius:5px; display:none; z-index:200;}

.overdue {background:#ffcccc;}

.pending {background:#fff3b0;}

.approved {background:#c6f6d5;}

.rejected {background:#d3d3d3;}

@media(max-width:600px){.search-add{flex-direction:column;} .search-add button{margin-top:10px;}}

</style>

</head>

<body>

<!-- LOGIN PAGE -->

<div id="login-page" class="page">

<h2>Library Login</h2>

<div class="input-group">

<input type="text" id="username" placeholder=" " required>

<label>Username</label>

</div>

<div class="input-group">

<input type="password" id="password" placeholder=" " required>

<label>Password</label>

</div>

<button onclick="login()">Login</button>

<p>Don't have an account? <span onclick="showRegister()">Register</span></p>

</div>

<!-- REGISTER PAGE -->

<div id="register-page" class="page" style="display:none;">

<h2>Register</h2>

<div class="input-group">

<input type="text" id="reg-username" placeholder=" " required>

<label>Username</label>

</div>

<div class="input-group">

<input type="password" id="reg-password" placeholder=" " required>

<label>Password</label>

</div>

<button onclick="register()">Register</button>

<p>Already have an account? <span onclick="showLogin()">Login</span></p>

</div>

<!-- DASHBOARD -->

<div id="dashboard" class="page" style="display:none;">

<div class="dashboard-header">

<h2>Welcome, <span id="user-display"></span></h2>

<button onclick="logout()">Logout</button>

</div>

<div class="tabs">

<button class="tab-btn active" onclick="showTab('books-tab')">Books</button>

<button class="tab-btn" onclick="showTab('requests-tab')">Requests</button>

</div>

<!-- Books Tab -->

<div id="books-tab" class="tab-content">

<div class="search-add">

<input type="text" id="search" placeholder="Search books..." onkeyup="displayBooks()">

<button id="add-book-btn" onclick="openModal(false)">Add Book</button>

</div>

<table id="books-table">

<thead>

<tr>

<th onclick="sortBooks('title')">Title &#8597;</th>

<th onclick="sortBooks('author')">Author &#8597;</th>

<th>Due Date</th>

<th>Action</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

<!-- Requests Tab -->

<div id="requests-tab" class="tab-content" style="display:none;">

<h3>Requests</h3>

<table id="requests-table">

<thead>

<tr>

<th>User</th>

<th>Book</th>

<th>Due Date</th>

<th>Status</th>

<th>Action</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

</div>

<!-- MODAL -->

<div id="book-modal" class="modal">

<div class="modal-content">

<span class="close" onclick="closeModal()">&times;</span>

<h3 id="modal-title">Add Book</h3>

<input type="text" id="modal-book-title" placeholder="Book Title">

<input type="text" id="modal-book-author" placeholder="Author">

<input type="date" id="modal-due-date">

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

</div>

</div>

<!-- NOTIFICATION -->

<div id="notification" class="notification"></div>

<script>

// INITIALIZATION

if(!localStorage.getItem('users')) localStorage.setItem('users',JSON.stringify([{username:'admin',password:'admin',role:'admin'}]));

if(!localStorage.getItem('books')) localStorage.setItem('books',JSON.stringify([]));

if(!localStorage.getItem('requests')) localStorage.setItem('requests',JSON.stringify([]));

let currentUser=null, editingIndex=null, currentSort={key:'',asc:true};

// LOGIN & REGISTER

function login(){

const u=document.getElementById('username').value;

const p=document.getElementById('password').value;

const users=JSON.parse(localStorage.getItem('users'));

const user=users.find(x=>x.username===u&&x.password===p);

if(!user) return alert('Invalid credentials!');

currentUser=user;

showDashboard();

}

function register(){

const u=document.getElementById('reg-username').value;

const p=document.getElementById('reg-password').value;

let users=JSON.parse(localStorage.getItem('users'));

if(users.find(x=>x.username===u)) return alert('Username exists!');

users.push({username:u,password:p,role:'user'});

localStorage.setItem('users',JSON.stringify(users));

alert('Registered successfully!');

showLogin();

}

function logout(){

currentUser=null;

document.getElementById('dashboard').style.display='none';

document.getElementById('login-page').style.display='block';

}

function showRegister(){document.getElementById('login-page').style.display='none'; document.getElementById('register-page').style.display='block';}

function showLogin(){document.getElementById('register-page').style.display='none'; document.getElementById('login-page').style.display='block';}

// DASHBOARD & TABS

function showDashboard(){

document.getElementById('login-page').style.display='none';

document.getElementById('register-page').style.display='none';

document.getElementById('dashboard').style.display='block';

document.getElementById('user-display').innerText=currentUser.username;

displayBooks();

displayRequests();

}

function showTab(tabId){

document.querySelectorAll('.tab-content').forEach(t=>t.style.display='none');

document.getElementById(tabId).style.display='block';

document.querySelectorAll('.tab-btn').forEach(b=>b.classList.remove('active'));

document.querySelector(`[onclick="showTab('${tabId}')"]`).classList.add('active');

}

// BOOKS

function getBooks(){return JSON.parse(localStorage.getItem('books'));}

function saveBooks(books){localStorage.setItem('books',JSON.stringify(books)); displayBooks();}

function openModal(isEdit=false,index=null){

document.getElementById('book-modal').style.display='block';

if(isEdit){

editingIndex=index;

const b=getBooks()[index];

document.getElementById('modal-title').innerText='Edit Book';

document.getElementById('modal-book-title').value=b.title;

document.getElementById('modal-book-author').value=b.author;

document.getElementById('modal-due-date').value=b.dueDate||'';

document.getElementById('modal-save-btn').innerText='Update';

} else{

editingIndex=null;

document.getElementById('modal-title').innerText='Add Book';

document.getElementById('modal-book-title').value='';

document.getElementById('modal-book-author').value='';

document.getElementById('modal-due-date').value='';

document.getElementById('modal-save-btn').innerText='Save';

}

}

function closeModal(){document.getElementById('book-modal').style.display='none';}

document.getElementById('modal-save-btn').onclick=function(){

const t=document.getElementById('modal-book-title').value;

const a=document.getElementById('modal-book-author').value;

const d=document.getElementById('modal-due-date').value;

if(!t||!a) return alert('Fill all fields');

let books=getBooks();

if(editingIndex!==null) books[editingIndex]={title:t,author:a,dueDate:d};

else books.push({title:t,author:a,dueDate:d});

saveBooks(books);

closeModal();

}

function deleteBook(index){let books=getBooks(); books.splice(index,1); saveBooks(books);}

// DISPLAY BOOKS

function displayBooks(){

const tbody=document.querySelector('#books-table tbody'); tbody.innerHTML='';

let books=getBooks();

const search=document.getElementById('search').value.toLowerCase();

books=books.filter(b=>b.title.toLowerCase().includes(search)||b.author.toLowerCase().includes(search));

if(currentSort.key) books.sort((a,b)=>currentSort.asc?a[currentSort.key].localeCompare(b[currentSort.key]):b[currentSort.key].localeCompare(a[currentSort.key]));

books.forEach((b,i)=>{

let today=(new Date()).toISOString().split('T')[0];

let rowClass=b.dueDate&&b.dueDate<today?'overdue':'';

let actions=currentUser.role==='admin'?`<button onclick="openModal(true,${i})">Edit</button><button onclick="deleteBook(${i})">Delete</button>`:`<button onclick="requestBook('${b.title}','${b.dueDate}')">Request</button>`;

tbody.innerHTML+=`<tr class="${rowClass}"><td>${b.title}</td><td>${b.author}</td><td>${b.dueDate||''}</td><td>${actions}</td></tr>`;

});

showOverdueNotification();

}

function sortBooks(key){currentSort.asc=currentSort.key===key?!currentSort.asc:true; currentSort.key=key; displayBooks();}

// REQUESTS

function getRequests(){return JSON.parse(localStorage.getItem('requests'));}

function saveRequests(r){localStorage.setItem('requests',JSON.stringify(r)); displayRequests();}

function requestBook(title,dueDate){let r=getRequests(); r.push({user:currentUser.username,book:title,status:'Pending',dueDate:dueDate||''}); saveRequests(r); alert('Requested successfully');}

function displayRequests(){

const tbody=document.querySelector('#requests-table tbody'); tbody.innerHTML='';

let r=getRequests();

if(currentUser.role!=='admin') r=r.filter(x=>x.user===currentUser.username);

r.forEach((req,i)=>{

let rowClass=req.status==='Pending'?'pending':req.status==='Approved'?'approved':req.status==='Rejected'?'rejected':'';

let actions=currentUser.role==='admin'&&req.status==='Pending'?`<button onclick="approve(${i})">Approve</button><button onclick="reject(${i})">Reject</button>`:'';

tbody.innerHTML+=`<tr class="${rowClass}"><td>${req.user}</td><td>${req.book}</td><td>${req.dueDate||''}</td><td>${req.status}</td><td>${actions}</td></tr>`;

});

}

function approve(i){let r=getRequests(); r[i].status='Approved'; saveRequests(r);}

function reject(i){let r=getRequests(); r[i].status='Rejected'; saveRequests(r);}

// OVERDUE NOTIFICATION

function showOverdueNotification(){if(currentUser.role!=='user') return; let overdue=getBooks().filter(b=>b.dueDate&&b.dueDate<(new Date()).toISOString().split('T')[0]); const notif=document.getElementById('notification'); if(overdue.length>0){ notif.innerText=`You have ${overdue.length} overdue book(s)!`; notif.style.display='block';} else notif.style.display='none';}

</script>

</body>

</html>