<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

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

<title>Combined Portal — Complaints & Student Management</title>

<style>

:root{--accent:#e23744;--bg:#181818;--card:#232323;--muted:#ccc}

body{font-family:Poppins,Arial,Helvetica,sans-serif;background:linear-gradient(135deg,#0f0f0f,#232323);color:#fff;margin:0;min-height:100vh;display:flex;align-items:center;justify-content:center;padding:20px}

.wrap{width:100%;max-width:1100px;background:var(--bg);border-radius:12px;padding:28px;box-shadow:0 8px 40px rgba(0,0,0,.6);position:relative}

h1,h2{color:var(--accent);margin:0 0 16px 0;text-transform:uppercase;letter-spacing:1px}

.row{display:flex;gap:16px}

.col{flex:1}

.small{font-size:.9rem;color:#bbb}

.notice{background:#222;color:#ffd;padding:10px;border-radius:8px;margin-bottom:12px;display:none}

label{display:block;margin-top:12px;font-weight:600;color:#ddd}

input,select,textarea,button{width:100%;padding:12px;border-radius:8px;border:none;background:#252525;color:#fff;font-size:1rem;margin-top:8px}

button{cursor:pointer}

button.ghost{background:none;border:1.5px solid var(--accent);color:var(--accent);padding:8px 12px;width:auto}

table{width:100%;border-collapse:collapse;margin-top:16px}

th,td{padding:10px;border:1px solid #444;text-align:left}

th{background:var(--accent);color:#fff}

.hidden{display:none}

.nav{display:flex;gap:12px;flex-wrap:wrap;margin-bottom:16px}

.pill{padding:8px 12px;border-radius:999px;background:#111;border:1px solid #333;cursor:pointer}

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

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

.title-row{display:flex;justify-content:space-between;align-items:center;margin-bottom:8px}

.center{display:flex;justify-content:center;align-items:center}

</style>

</head>

<body>

<div class="wrap" id="appWrap">

<div id="envNotice" class="notice"></div>

<div id="loginScreen">

<div class="title-row">

<h1>Unified Portal</h1>

<div class="small">Complaint Portal + Student Management</div>

</div>

<div class="small" style="margin-bottom:10px">Select role and optionally choose which module you want after login (admin sees both).</div>

<form id="loginForm">

<label for="role">Role</label>

<select id="role">

<option value="customer">Customer</option>

<option value="admin">Admin</option>

</select>

<label for="email">Email</label>

<input id="email" type="email" placeholder="Enter email" required/>

<label for="password">Password (for admin if using Firebase Auth)</label>

<input id="password" type="password" placeholder="Password (optional)"/>

<div style="margin-top:14px" class="flex-row">

<button class="ghost" type="button" id="demoAdmin">Use demo admin (admin@gmail.com)</button>

<div style="flex:1"></div>

<button type="submit">Login</button>

</div>

<p class="small" style="margin-top:10px">Note: This single-file app runs in local mode by default using localStorage. It also contains optional Firebase initialization (paste config in script to enable).</p>

</form>

</div>

<div id="mainScreen" class="hidden">

<div class="title-row">

<h1 id="mainTitle">Dashboard</h1>

<div class="flex-row">

<div id="currentUser" class="small" style="margin-right:12px"></div>

<button class="ghost" id="logoutBtn">Logout</button>

</div>

</div>

<div class="nav" id="moduleNav">

<div class="pill" data-module="complaint" tabindex="0">Complaint Portal</div>

<div class="pill" data-module="student" tabindex="0">Student Management</div>

</div>

<div id="contentArea">

<!-- Complaint Portal -->

<div id="complaintModule" class="module">

<div class="row">

<div class="col">

<div class="container-card">

<h2>Submit Complaint</h2>

<form id="complaintForm">

<label>Order ID</label>

<input id="c\_orderId" type="text" required />

<label>Issue Type</label>

<select id="c\_issueType">

<option>Late Delivery</option>

<option>Wrong Order</option>

<option>Food Quality</option>

<option>Refund Issue</option>

<option>Other</option>

</select>

<label>Description</label>

<textarea id="c\_description" rows="4" required></textarea>

<button type="submit">Submit Complaint</button>

</form>

<p id="complaintSuccess" class="small hidden" style="color:#4caf50;margin-top:10px">Complaint submitted successfully!</p>

</div>

</div>

<div class="col">

<div class="container-card">

<h2>Admin Complaints</h2>

<table id="complaintTable">

<thead>

<tr><th>ID</th><th>Order ID</th><th>Issue</th><th>Status</th><th>Action</th></tr>

</thead>

<tbody id="complaintList"><tr><td colspan="5" class="center">No complaints yet.</td></tr></tbody>

</table>

</div>

</div>

</div>

</div>

<!-- Student Management Module -->

<div id="studentModule" class="module hidden">

<div class="title-row">

<h2>Student Management</h2>

<div class="actions">

<button id="showAddStudent" class="ghost">Add Student</button>

<button id="refreshStudents" class="ghost">Refresh</button>

</div>

</div>

<div id="studentForms" class="hidden">

<form id="addStudentForm" style="margin-bottom:10px">

<label>Roll</label>

<input id="s\_roll" required />

<label>Name</label>

<input id="s\_name" required />

<label>Email</label>

<input id="s\_email" type="email" required />

<label>Course</label>

<input id="s\_course" required />

<div style="display:flex;gap:8px;margin-top:10px">

<button type="submit">Save Student</button>

<button type="button" id="cancelAdd" class="ghost">Cancel</button>

</div>

</form>

<form id="editStudentForm" class="hidden" style="margin-bottom:10px">

<input id="s\_edit\_id" type="hidden" />

<label>Roll</label>

<input id="s\_edit\_roll" required />

<label>Name</label>

<input id="s\_edit\_name" required />

<label>Email</label>

<input id="s\_edit\_email" type="email" required />

<label>Course</label>

<input id="s\_edit\_course" required />

<div style="display:flex;gap:8px;margin-top:10px">

<button type="submit">Update Student</button>

<button type="button" id="cancelEdit" class="ghost">Cancel</button>

</div>

</form>

</div>

<table id="studentTable">

<thead><tr><th>ID</th><th>Roll</th><th>Name</th><th>Email</th><th>Course</th><th>Actions</th></tr></thead>

<tbody id="studentList"><tr><td colspan="6" class="center">No students yet.</td></tr></tbody>

</table>

</div>

<!-- Delivery Partner Page (appears after resolve) -->

<div id="deliveryPage" class="module hidden">

<h2>Resolved — Delivery Partner Details</h2>

<div id="deliveryInfo"></div>

<div style="margin-top:14px">

<button id="backToComplaints" class="ghost">Back to Complaints</button>

</div>

</div>

</div>

</div>

</div>

<script>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Single-file Combined Portal:

\* - Complaint Portal (customer + admin)

\* - Student Management (CRUD)

\* - All data stored in localStorage under keys: 'complaints' and 'students'

\* - Optional Firebase placeholders preserved but disabled by default

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* ---------- Firebase placeholder (not active by default) ---------- \*/

const firebaseConfig = {

apiKey: "YOUR\_API\_KEY",

authDomain: "YOUR\_PROJECT\_ID.firebaseapp.com",

projectId: "YOUR\_PROJECT\_ID",

storageBucket: "YOUR\_PROJECT\_ID.appspot.com",

messagingSenderId: "YOUR\_SENDER\_ID",

appId: "YOUR\_APP\_ID"

};

function hasValidConfig(c){ if(!c||typeof c!=='object')return false; const keys=['apiKey','authDomain','projectId','appId']; return keys.every(k=>c[k] && !c[k].includes('YOUR\_')) }

/\* ---------- DOM refs ---------- \*/

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

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

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

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

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

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

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

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

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

/\* ---------- Auth & UI state ---------- \*/

let role = null;

let email = null;

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

document.getElementById('role').value = 'admin';

document.getElementById('email').value = 'admin@gmail.com';

document.getElementById('password').value = 'Admin@123';

});

document.getElementById('loginForm').addEventListener('submit', (e) => {

e.preventDefault();

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

email = document.getElementById('email').value.trim().toLowerCase();

// simple local admin check if firebase not configured

if (role === 'admin' && !hasValidConfig(firebaseConfig) && email !== 'admin@gmail.com') {

alert('In local mode only admin@gmail.com is allowed as admin. Use demo admin button or change email.');

return;

}

loginScreen.classList.add('hidden');

mainScreen.classList.remove('hidden');

currentUser.textContent = `${email} (${role})`;

// default module

showModule('complaint');

// load lists

loadComplaints();

loadStudents();

});

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

role = null; email = null;

mainScreen.classList.add('hidden');

loginScreen.classList.remove('hidden');

// reset forms visibility

document.getElementById('studentForms').classList.add('hidden');

document.getElementById('addStudentForm').reset();

document.getElementById('editStudentForm').reset();

});

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

const el = ev.target.closest('.pill');

if (!el) return;

const m = el.getAttribute('data-module');

showModule(m);

});

function showModule(name) {

[complaintModule, studentModule, deliveryPage].forEach(n => n.classList.add('hidden'));

if (name === 'complaint') complaintModule.classList.remove('hidden');

if (name === 'student') studentModule.classList.remove('hidden');

if (name === 'delivery') deliveryPage.classList.remove('hidden');

}

/\* ---------- Complaints (localStorage) ---------- \*/

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

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

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

complaintForm.addEventListener('submit', (e) => {

e.preventDefault();

const orderId = document.getElementById('c\_orderId').value.trim();

const issueType = document.getElementById('c\_issueType').value;

const description = document.getElementById('c\_description').value.trim();

if (!orderId || !description) { alert('Please fill required fields'); return; }

const complaints = JSON.parse(localStorage.getItem('complaints')) || [];

const complaint = { id: Date.now(), orderId, issueType, description, status:'Pending', createdAt: new Date().toISOString() };

complaints.push(complaint);

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

complaintSuccess.classList.remove('hidden');

setTimeout(()=>{ complaintSuccess.classList.add('hidden'); clearComplaintForm(); }, 1400);

loadComplaints();

});

function clearComplaintForm(){

document.getElementById('c\_orderId').value = '';

document.getElementById('c\_description').value = '';

}

function loadComplaints(){

const complaints = JSON.parse(localStorage.getItem('complaints')) || [];

if (!Array.isArray(complaints) || complaints.length === 0) {

complaintList.innerHTML = '<tr><td colspan="5" class="center">No complaints yet.</td></tr>';

return;

}

complaintList.innerHTML = complaints.map(c => {

return `<tr>

<td>${c.id}</td>

<td>${escapeHtml(c.orderId)}</td>

<td>${escapeHtml(c.issueType)}</td>

<td>${escapeHtml(c.status)}</td>

<td><button class="ghost" onclick="resolveComplaint('${c.id}')">Resolve</button></td>

</tr>`;

}).join('');

}

window.resolveComplaint = function(id){

// update status

try {

const complaints = JSON.parse(localStorage.getItem('complaints')) || [];

const updated = complaints.map(c => String(c.id) === String(id) ? {...c, status:'Resolved'} : c);

localStorage.setItem('complaints', JSON.stringify(updated));

} catch(e){ console.error('resolve error',e); alert('Failed to resolve locally'); return; }

// show delivery details

showDelivery();

};

function showDelivery(){

const names = ['Ravi Kumar','Amit Sharma','Rajesh Singh','Suresh Yadav','Deepak Patel','Anil Mehta','Vikas Rao','Manish Verma'];

const vehicles = ['MH12AB1234','DL3CAF4567','KA05MN6789','TN10XY4321','GJ18JK9087','WB20CD2345','PB10EF6789'];

const name = names[Math.floor(Math.random()\*names.length)];

const phone = '+91' + (Math.floor(6000000000 + Math.random()\*3999999999)).toString();

const vehicle = vehicles[Math.floor(Math.random()\*vehicles.length)];

const arrival = new Date(Date.now() + Math.floor(Math.random()\*30+1)\*60000).toLocaleTimeString('en-IN',{hour:'2-digit',minute:'2-digit'});

deliveryInfo.innerHTML = `

<p><strong>Driver Name:</strong> ${name}</p>

<p><strong>Phone:</strong> ${phone}</p>

<p><strong>Vehicle Number:</strong> ${vehicle}</p>

<p><strong>Arrival Time:</strong> ${arrival}</p>

`;

showModule('delivery');

}

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

showModule('complaint');

loadComplaints();

});

/\* ---------- Student Management (localStorage) ---------- \*/

const studentListEl = document.getElementById('studentList');

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

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

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

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

const showAddStudentBtn = document.getElementById('showAddStudent');

const refreshStudentsBtn = document.getElementById('refreshStudents');

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

studentForms.classList.remove('hidden');

addStudentForm.classList.remove('hidden');

editStudentForm.classList.add('hidden');

addStudentForm.reset();

document.getElementById('s\_roll').focus();

});

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

studentForms.classList.add('hidden');

});

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

editStudentForm.classList.add('hidden');

});

addStudentForm.addEventListener('submit', (e) => {

e.preventDefault();

const roll = document.getElementById('s\_roll').value.trim();

const name = document.getElementById('s\_name').value.trim();

const emailS = document.getElementById('s\_email').value.trim();

const course = document.getElementById('s\_course').value.trim();

if(!roll||!name||!emailS||!course){ alert('Fill all fields'); return; }

const students = JSON.parse(localStorage.getItem('students')) || [];

const s = { id: Date.now(), roll, name, email: emailS, course };

students.push(s);

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

studentForms.classList.add('hidden');

loadStudents();

});

editStudentForm.addEventListener('submit', (e) => {

e.preventDefault();

const id = Number(document.getElementById('s\_edit\_id').value);

const roll = document.getElementById('s\_edit\_roll').value.trim();

const name = document.getElementById('s\_edit\_name').value.trim();

const emailS = document.getElementById('s\_edit\_email').value.trim();

const course = document.getElementById('s\_edit\_course').value.trim();

try{

const students = JSON.parse(localStorage.getItem('students')) || [];

const updated = students.map(st => st.id === id ? { ...st, roll, name, email: emailS, course } : st);

localStorage.setItem('students', JSON.stringify(updated));

editStudentForm.classList.add('hidden');

loadStudents();

}catch(e){ console.error(e); alert('Update failed'); }

});

refreshStudentsBtn.addEventListener('click', loadStudents);

function loadStudents(){

const students = JSON.parse(localStorage.getItem('students')) || [];

if (!Array.isArray(students) || students.length === 0){

studentListEl.innerHTML = '<tr><td colspan="6" class="center">No students yet.</td></tr>';

return;

}

studentListEl.innerHTML = students.map(s => {

return `<tr>

<td>${s.id}</td>

<td>${escapeHtml(s.roll)}</td>

<td>${escapeHtml(s.name)}</td>

<td>${escapeHtml(s.email)}</td>

<td>${escapeHtml(s.course)}</td>

<td>

<button class="ghost" onclick="editStudent('${s.id}')">Edit</button>

<button class="ghost" onclick="deleteStudent('${s.id}')">Delete</button>

</td>

</tr>`;

}).join('');

}

window.editStudent = function(id){

const students = JSON.parse(localStorage.getItem('students')) || [];

const st = students.find(s => String(s.id) === String(id));

if (!st) { alert('Student not found'); return; }

studentForms.classList.remove('hidden');

addStudentForm.classList.add('hidden');

editStudentForm.classList.remove('hidden');

document.getElementById('s\_edit\_id').value = st.id;

document.getElementById('s\_edit\_roll').value = st.roll;

document.getElementById('s\_edit\_name').value = st.name;

document.getElementById('s\_edit\_email').value = st.email;

document.getElementById('s\_edit\_course').value = st.course;

};

window.deleteStudent = function(id){

if (!confirm('Delete this student?')) return;

try {

const students = JSON.parse(localStorage.getItem('students')) || [];

const filtered = students.filter(s => String(s.id) !== String(id));

localStorage.setItem('students', JSON.stringify(filtered));

loadStudents();

} catch(e){ console.error(e); alert('Delete failed'); }

};

/\* ---------- Utilities ---------- \*/

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

// Initialize demo data if none present (helpful for demo)

(function seedDemoData(){

if (!localStorage.getItem('students')){

const sample = [

{id: Date.now()+1, roll:'2023CS001', name:'Ajay Kumar', email:'ajay@example.com', course:'Computer Science'},

{id: Date.now()+2, roll:'2023ME002', name:'Sneha Patel', email:'sneha@example.com', course:'Mechanical'}

];

localStorage.setItem('students', JSON.stringify(sample));

}

if (!localStorage.getItem('complaints')){

const sampleC = [

{id: Date.now()+11, orderId:'ORD1001', issueType:'Late Delivery', description:'Food arrived cold', status:'Pending', createdAt: new Date().toISOString()}

];

localStorage.setItem('complaints', JSON.stringify(sampleC));

}

})();

// on load: nothing until login; but attach window.onerror to help display script issues

window.addEventListener('error', function(ev){

if (ev.message === 'Script error.' && ev.filename === '') {

console.error('A script error occurred. This often means a third-party script failed to load or network blocked resources.');

envNotice.style.display = 'block';

envNotice.textContent = 'Some resources failed to load; running in local mode.';

} else {

console.error('Error event', ev);

}

});

// Expose load functions in case needed from console

window.loadComplaints = loadComplaints;

window.loadStudents = loadStudents;

</script>

</body>

</html>