Case Study:

Product Management System (PMS)

1. **Project Overview Objective: Build a web-based Product Management System where an admin can manage products (CRUD operations) and users can browse products, add them to cart, and checkout.**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Product Management System (PMS)</title>
  <!-- Bootstrap 5 -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" />
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css"
rel="stylesheet" />
  <style>
    :root {
      --brand: #6f42c1;
      --brand-2: #e9d8fd;
    }
    body {
      background: linear-gradient(180deg, #f8f9fa 0%, #fff 100%);
      min-height: 100vh;
    }
    .navbar-brand {
      font-weight: 800;
      letter-spacing: .5px;
    }
    .brand-badge {
      background: var(--brand);
  color: #fff;
  border-radius: 999px;
  padding: .25rem .75rem;
  font-weight: 600;
}
.card {
  border: 0;
  border-radius: 1rem;
  box-shadow: 0 10px 24px rgba(0,0,0,.05);
}
.card-header {
  background: linear-gradient(135deg, var(--brand), #845ef7);
  color: #fff;
  border-top-left-radius: 1rem !important;
  border-top-right-radius: 1rem !important;
}
.btn-brand {
  background: var(--brand);
  color: #fff;
  border: none;
}
.btn-brand:hover { filter: brightness(1.05); }

.pill {
  background: var(--brand-2);
  color: #5224aa;
  border-radius: 999px;
  padding: .25rem .75rem;
  font-weight: 600;
  font-size: .85rem;
}
```

```
    .hidden { display: none !important; }

    /* Product image holder */
    .product-img {
      width: 100%;
      aspect-ratio: 16/10;
      object-fit: cover;
      border-radius: .75rem;
    }

    .pointer { cursor: pointer; }

    /* Small helper for required marker */
    .req::after{ content:" *"; color:#dc3545; }

    /* Footer */
    footer { color:#6c757d; }

    /* Compact table */
    .table-sm td, .table-sm th { padding: .45rem .5rem; }
  </style>
</head>
<body>
  <!-- NAVBAR (dynamic links render based on role/state) -->
  <nav class="navbar navbar-expand-lg bg-white border-bottom sticky-top">
    <div class="container">
      <a class="navbar-brand d-flex align-items-center gap-2" href="#" id="brandGoHome">
        <span class="brand-badge">PMS</span>
        <span>Product Management System</span>
      </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navContent">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navContent">
        <ul class="navbar-nav ms-auto align-items-lg-center gap-lg-2" id="navLinks">
        </ul>
      </div>
    </div>
  </nav>

  <main class="container py-4" id="appRoot">

    <!-- WELCOME -->
    <section id="view-welcome" class="mt-4">
      <div class="row justify-content-center">
        <div class="col-lg-8">
          <div class="card">
            <div class="card-header d-flex justify-content-between align-items-center">
              <h5 class="mb-0">Welcome to PMS</h5>
              <span class="pill">Simple • Clean • Fast</span>
            </div>
            <div class="card-body p-4">
              <div class="text-center">
                <img src="https://cdn.jsdelivr.net/gh/twitter/twemoji@14.0.2/assets/svg/1f6d2.svg" width="56" height="56" alt="cart"/>
                <h2 class="mt-3">Manage & Shop Products Seamlessly</h2>
                <p class="text-secondary">Admins can manage the catalog; users can browse, add to cart, and checkout.</p>
                <div class="d-flex gap-3 justify-content-center mt-4">
                  <button class="btn btn-brand px-4" id="gotoLogin">Login</button>
                  <button class="btn btn-outline-secondary px-4" id="gotoSignup">Sign Up</button>
```

```html
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

<!-- SIGNUP -->
<section id="view-signup" class="hidden mt-4">
  <div class="row justify-content-center">
    <div class="col-lg-6">
      <div class="card">
        <div class="card-header"><h5 class="mb-0">Create an account</h5></div>
        <div class="card-body">
          <form id="signupForm" class="row g-3">
            <div class="col-12">
              <label class="form-label req">Full Name</label>
              <input type="text" class="form-control" id="suName" required />
            </div>
            <div class="col-12">
              <label class="form-label req">Email</label>
              <input type="email" class="form-control" id="suEmail" required />
            </div>
            <div class="col-12">
              <label class="form-label req">Password</label>
              <input type="password" class="form-control" id="suPassword" minlength="4" required />
            </div>
            <div class="col-12">
              <label class="form-label req">Role</label>
              <select class="form-select" id="suRole" required>
```

```
                    <option value="user">User</option>
                    <option value="admin">Admin</option>
                  </select>
                </div>
                <div class="col-12 d-flex gap-2">
                  <button class="btn btn-brand" type="submit">Sign Up</button>
                  <button class="btn btn-outline-secondary" type="button" id="backToLoginFromSignup">Back to Login</button>
                </div>
                <div class="col-12 small text-secondary">Account data is stored in JSON Server (<code>/users</code>).</div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </section>

    <!-- LOGIN -->
    <section id="view-login" class="hidden mt-4">
      <div class="row justify-content-center">
        <div class="col-lg-5">
          <div class="card">
            <div class="card-header"><h5 class="mb-0">Login</h5></div>
            <div class="card-body">
              <form id="loginForm" class="row g-3">
                <div class="col-12">
                  <label class="form-label req">Email</label>
                  <input type="email" class="form-control" id="liEmail" required />
                </div>
                <div class="col-12">
                  <label class="form-label req">Password</label>
                  <input type="password" class="form-control" id="liPassword" required />
                </div>
                <div class="col-12 d-flex gap-2">
                  <button class="btn btn-brand" type="submit">Login</button>
                  <button class="btn btn-outline-secondary" type="button"
  id="backToWelcomeFromLogin">Cancel</button>
                </div>
                <div class="col-12 small text-secondary">We'll verify against
  <code>/users?email=&password=</code>.</div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </section>

    <!-- ADMIN DASHBOARD WRAPPER -->
    <section id="view-admin" class="hidden mt-3">
      <div class="row g-3">
        <!-- Add Product -->
        <div class="col-12 col-lg-5">
          <div class="card h-100">
            <div class="card-header d-flex align-items-center justify-content-between">
              <h6 class="mb-0">Add / Edit Product</h6>
              <span class="pill" id="apMode">New</span>
            </div>
            <div class="card-body">
              <form id="productForm" class="row g-3">
                <input type="hidden" id="prodId" />
                <div class="col-12">
                  <label class="form-label req">Product Name</label>
                  <input type="text" class="form-control" id="prodName" required />
          </div>
          <div class="col-12">
           <label class="form-label req">Description</label>
           <textarea class="form-control" id="prodDesc" rows="3" required></textarea>
          </div>
          <div class="col-12 col-md-6">
           <label class="form-label req">Price (₹)</label>
           <input type="number" min="0" class="form-control" id="prodPrice" required />
          </div>
          <div class="col-12">
           <label class="form-label">Image URL (optional)</label>
           <input type="url" class="form-control" id="prodImageUrl" placeholder="https://..." />
          </div>
          <div class="col-12">
           <label class="form-label">Or Upload Image (optional)</label>
           <input type="file" class="form-control" id="prodImageFile" accept="image/*" />
           <div class="form-text">Files are converted to data URLs and saved inline.</div>
          </div>
          <div class="col-12 d-flex gap-2">
           <button class="btn btn-brand" type="submit" id="saveProductBtn">Save
  Product</button>
           <button class="btn btn-outline-secondary" type="button"
  id="resetProductForm">Reset</button>
          </div>
         </form>
        </div>
       </div>
      </div>
      <!-- Product List -->
      <div class="col-12 col-lg-7">
       <div class="card h-100">
        <div class="card-header d-flex align-items-center justify-content-between">
```

```
              <h6 class="mb-0">Product List</h6>
              <input type="search" class="form-control form-control-sm w-auto" id="adminSearch" placeholder="Search..." />
            </div>
            <div class="card-body">
              <div id="adminProducts" class="row g-3">
                <!-- Admin product cards injected here -->
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>

    <!-- USER DASHBOARD WRAPPER -->
    <section id="view-user" class="hidden mt-3">
      <div class="row g-3">
        <div class="col-12">
          <div class="card">
            <div class="card-header d-flex align-items-center justify-content-between">
              <h6 class="mb-0">Browse Products</h6>
              <input type="search" class="form-control form-control-sm w-auto" id="userSearch" placeholder="Search products..." />
            </div>
            <div class="card-body">
              <div id="userProducts" class="row g-3">
                <!-- User product cards injected here -->
              </div>
            </div>
          </div>
        </div>
      </div>
```

```
</section>


<!-- CART / CHECKOUT -->
<section id="view-cart" class="hidden mt-3">
 <div class="row g-3">
  <div class="col-12">
   <div class="card">
    <div class="card-header d-flex align-items-center justify-content-between">
     <h6 class="mb-0">Your Cart</h6>
     <span class="pill" id="cartCountPill">0 items</span>
    </div>
    <div class="card-body">
     <div class="table-responsive">
      <table class="table table-sm align-middle" id="cartTable">
       <thead class="table-light">
        <tr>
         <th>#</th>
         <th>Product</th>
         <th class="text-end">Price (₹)</th>
         <th class="text-center">Action</th>
        </tr>
       </thead>
       <tbody></tbody>
       <tfoot>
        <tr>
         <th colspan="2" class="text-end">Total</th>
         <th class="text-end" id="cartTotal">₹0</th>
         <th></th>
        </tr>
       </tfoot>
      </table>
```

```
       </div>
       <div class="d-flex gap-2">
        <button class="btn btn-brand" id="placeOrderBtn"><i class="bi bi-bag-check"></i> Place Order</button>
        <button class="btn btn-outline-secondary" id="continueShoppingBtn">Continue Shopping</button>
       </div>
       <small class="text-secondary d-block mt-2">Cart is saved in <code>localStorage</code> per user.</small>
      </div>
     </div>
    </div>
   </div>
  </section>

 </main>

 <footer class="container py-4">
  <div class="d-flex justify-content-between small">
   <div>© <span id="year"></span> PMS • Built with Bootstrap 5</div>
   <div>Demo backend: JSON Server on <code>http://localhost:3000</code></div>
  </div>
 </footer>

 <script>
  // ====== CONFIG ======
  const BASE_URL = 'http://localhost:3000';

  // ====== UTILITIES ======
  const qs = (sel) => document.querySelector(sel);
  const qsa = (sel) => [...document.querySelectorAll(sel)];
```

```
    function showView(id){
      ['#view-welcome','#view-signup','#view-login','#view-admin','#view-user','#view-cart']
        .forEach(v => qs(v).classList.add('hidden'));
      qs(id).classList.remove('hidden');
      window.scrollTo({ top: 0, behavior: 'smooth' });
    }

    function toast(msg, type = 'info'){
      // Simple inline toast
      const box = document.createElement('div');
      box.className = `position-fixed top-0 start-50 translate-middle-x mt-3 z-3`;
      box.innerHTML = `<div class="alert alert-${type} shadow">${msg}</div>`;
      document.body.appendChild(box);
      setTimeout(()=> box.remove(), 2200);
    }

    function money(n){
      const val = Number(n || 0);
      return `₹${val.toLocaleString('en-IN', { maximumFractionDigits: 2 })}`;
    }

    function setNavLinks(){
      const nav = qs('#navLinks');
      nav.innerHTML = '';
      const user = getSession();

      const li = (html) => { const e = document.createElement('li'); e.className='nav-item';
  e.innerHTML = html; return e; };

      if(!user){
        nav.append(
          li(`<a class="nav-link" href="#" id="navWelcome">Welcome</a>`),
          li(`<a class="nav-link" href="#" id="navLogin">Login</a>`),
          li(`<a class="nav-link" href="#" id="navSignup">Sign Up</a>`),
        );
      } else if(user.role === 'admin'){
        nav.append(
          li(`<a class="nav-link" href="#" id="navAdminDashboard">Dashboard</a>`),
          li(`<a class="nav-link" href="#" id="navLogout">Logout</a>`),
          li(`<span class="navbar-text ms-2 pill"><i class="bi bi-person-badge"></i> ${user.name}</span>`)
        );
      } else {
        const count = getCart().length;
        nav.append(
          li(`<a class="nav-link" href="#" id="navUserProducts">Products</a>`),
          li(`<a class="nav-link position-relative" href="#" id="navCart">Cart
                <span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-danger" id="navCartCount">${count}</span>
              </a>`),
          li(`<a class="nav-link" href="#" id="navCheckout">Checkout</a>`),
          li(`<a class="nav-link" href="#" id="navLogout">Logout</a>`),
          li(`<span class="navbar-text ms-2 pill"><i class="bi bi-person"></i> ${user.name}</span>`)
        );
      }

      // Bind nav events freshly each time
      qs('#navWelcome')?.addEventListener('click', ()=> showView('#view-welcome'));
      qs('#navLogin')?.addEventListener('click', ()=> showView('#view-login'));
      qs('#navSignup')?.addEventListener('click', ()=> showView('#view-signup'));
      qs('#navAdminDashboard')?.addEventListener('click', ()=> { renderAdmin(); showView('#view-admin'); });
      qs('#navUserProducts')?.addEventListener('click', ()=> { renderUserProducts(); showView('#view-user'); });
      qs('#navCart')?.addEventListener('click', ()=> { renderCart(); showView('#view-cart'); });
      qs('#navCheckout')?.addEventListener('click', ()=> { renderCart(); showView('#view-cart'); });
      qs('#navLogout')?.addEventListener('click', logout);
    }

    function setYear(){ qs('#year').textContent = new Date().getFullYear(); }

    // ====== SESSION & CART ======
    function getSession(){
      try { return JSON.parse(localStorage.getItem('pms_user')); } catch { return null; }
    }
    function setSession(user){ localStorage.setItem('pms_user', JSON.stringify(user)); setNavLinks(); }
    function clearSession(){ localStorage.removeItem('pms_user'); setNavLinks(); }

    function cartKey(){ const u = getSession(); return u ? `pms_cart_${u.id}` : 'pms_cart'; }
    function getCart(){ try { return JSON.parse(localStorage.getItem(cartKey())) || []; } catch { return []; } }
    function setCart(items){ localStorage.setItem(cartKey(), JSON.stringify(items)); updateCartBadge(); }
    function clearCart(){ localStorage.removeItem(cartKey()); updateCartBadge(); }
    function updateCartBadge(){ const c = qs('#navCartCount'); if(c) c.textContent = getCart().length; }

    // ====== API HELPERS (JSON Server) ======
    async function api(url, opts={}){
      const res = await fetch(url, { headers: { 'Content-Type': 'application/json' }, ...opts });
      if(!res.ok) throw new Error(`API ${res.status}: ${await res.text()}`);
      return res.json();
    }

    // ====== AUTH ======
    async function handleSignup(e){
      e.preventDefault();
      const name = qs('#suName').value.trim();
      const email = qs('#suEmail').value.trim().toLowerCase();
      const password = qs('#suPassword').value;
      const role = qs('#suRole').value;
      if(!name || !email || !password) return toast('Please fill all required fields', 'warning');

      // prevent duplicate email
      const existing = await api(`${BASE_URL}/users?email=${encodeURIComponent(email)}`);
      if(existing.length){ toast('Email already registered.', 'danger'); return; }

      const user = await api(`${BASE_URL}/users`, {
        method: 'POST',
        body: JSON.stringify({ name, email, password, role })
      });
      toast('Signup successful! Please login.', 'success');
      showView('#view-login');
      qs('#liEmail').value = user.email;
    }

    async function handleLogin(e){
      e.preventDefault();
      const email = qs('#liEmail').value.trim().toLowerCase();
      const password = qs('#liPassword').value;
      const users = await
  api(`${BASE_URL}/users?email=${encodeURIComponent(email)}&password=${encodeURIComponen
  t(password)}`);
      if(!users.length){ toast('Invalid credentials', 'danger'); return; }
      const user = users[0];
      setSession(user);
      toast(`Welcome, ${user.name}!`, 'success');
      if(user.role === 'admin'){ renderAdmin(); showView('#view-admin'); }
      else { renderUserProducts(); showView('#view-user'); }
    }

    function logout(){
      clearSession();
      showView('#view-welcome');
      toast('Logged out. See you again!', 'secondary');
    }

    // ====== ADMIN: PRODUCTS CRUD ======
    async function renderAdmin(){
      const listWrap = qs('#adminProducts');
      listWrap.innerHTML = '<div class="text-secondary">Loading...</div>';
      const products = await api(`${BASE_URL}/products`);
      const term = qs('#adminSearch').value?.toLowerCase?.() || '';
      const data = products.filter(p => [p.name,p.description].join(' ').toLowerCase().includes(term));
      if(!data.length){ listWrap.innerHTML = '<p class="text-secondary">No products yet.</p>'; return; }
      listWrap.innerHTML = data.map(p => `
        <div class="col-12 col-md-6">
          <div class="card h-100">
            <div class="card-body">
              <img class="product-img mb-3" src="${p.image || 'https://picsum.photos/seed/prod'+p.id+'/640/360'}" alt="${p.name}"/>
              <h5>${p.name}</h5>
              <p class="text-secondary small mb-2">${p.description}</p>
              <div class="d-flex justify-content-between align-items-center">
                <strong>${money(p.price)}</strong>
                <div class="btn-group">
                  <button class="btn btn-sm btn-outline-primary" data-action="edit" data-id="${p.id}"><i class="bi bi-pencil"></i></button>
```

```
         <button class="btn btn-sm btn-outline-danger" data-action="delete" data-id="${p.id}"><i class="bi bi-trash"></i></button>
        </div>
       </div>
      </div>
     </div>
    </div>`).join('');

   // Bind buttons
   listWrap.querySelectorAll('[data-action="edit"]').forEach(btn => btn.addEventListener('click', ()=> loadProductForEdit(btn.dataset.id)));
   listWrap.querySelectorAll('[data-action="delete"]').forEach(btn => btn.addEventListener('click', ()=> deleteProduct(btn.dataset.id)));
  }

  function resetProductForm(){
   qs('#prodId').value = '';
   qs('#prodName').value = '';
   qs('#prodDesc').value = '';
   qs('#prodPrice').value = '';
   qs('#prodImageUrl').value = '';
   qs('#prodImageFile').value = '';
   qs('#apMode').textContent = 'New';
  }

  async function loadProductForEdit(id){
   const p = await api(`${BASE_URL}/products/${id}`);
   qs('#prodId').value = p.id;
   qs('#prodName').value = p.name;
   qs('#prodDesc').value = p.description;
   qs('#prodPrice').value = p.price;
   qs('#prodImageUrl').value = p.image || '';
```

```
      qs('#prodImageFile').value = '';
      qs('#apMode').textContent = 'Edit';
      toast('Loaded for editing', 'info');
    }

    function getImagePromise(){
      return new Promise((resolve)=>{
        const file = qs('#prodImageFile').files?.[0];
        const url = qs('#prodImageUrl').value.trim();
        if(file){
          const reader = new FileReader();
          reader.onload = () => resolve(reader.result);
          reader.readAsDataURL(file);
        } else if(url){ resolve(url); }
        else resolve('');
      });
    }

    async function handleProductSave(e){
      e.preventDefault();
      const id = qs('#prodId').value;
      const name = qs('#prodName').value.trim();
      const description = qs('#prodDesc').value.trim();
      const price = Number(qs('#prodPrice').value);
      const image = await getImagePromise();

      if(!name || !description || isNaN(price)) return toast('Please fill all required product fields', 'warning');

      const payload = { name, description, price, image };
```

```
   if(id){ // update
    await api(`${BASE_URL}/products/${id}`, { method: 'PUT', body: JSON.stringify({ id: Number(id),
...payload })});
    toast('Product updated', 'success');
   } else { // create
    await api(`${BASE_URL}/products`, { method: 'POST', body: JSON.stringify(payload)});
    toast('Product created', 'success');
   }

   resetProductForm();
   renderAdmin();
  }

  async function deleteProduct(id){
   if(!confirm('Delete this product?')) return;
   await api(`${BASE_URL}/products/${id}`, { method: 'DELETE' });
   toast('Product deleted', 'secondary');
   renderAdmin();
  }

  // ====== USER: BROWSE & CART ======
  async function renderUserProducts(){
   const wrap = qs('#userProducts');
   wrap.innerHTML = '<div class="text-secondary">Loading...</div>';
   const products = await api(`${BASE_URL}/products`);
   const term = qs('#userSearch').value?.toLowerCase?.() || '';
   const data = products.filter(p => [p.name,p.description].join(' ').toLowerCase().includes(term));
   if(!data.length){ wrap.innerHTML = '<p class="text-secondary">No products found.</p>'; return; }
   wrap.innerHTML = data.map(p => `
    <div class="col-12 col-sm-6 col-lg-4">
     <div class="card h-100">
```

```
      <div class="card-body d-flex flex-column">
       <img class="product-img mb-3" src="${p.image || 'https://picsum.photos/seed/prod'+p.id+'/640/360'}" alt="${p.name}"/>
       <h6>${p.name}</h6>
       <p class="text-secondary small">${p.description}</p>
       <div class="mt-auto d-flex justify-content-between align-items-center">
        <strong>${money(p.price)}</strong>
        <button class="btn btn-sm btn-brand" data-add-cart="${p.id}"><i class="bi bi-cart-plus"></i> Add</button>
       </div>
      </div>
     </div>
    </div>`).join('');

   wrap.querySelectorAll('[data-add-cart]').forEach(btn => btn.addEventListener('click', ()=> addToCart(Number(btn.dataset.addCart))));
  }

  function addToCart(productId){
   const items = getCart();
   items.push(productId);
   setCart(items);
   toast('Added to cart', 'success');
   updateCartBadge();
  }

  async function renderCart(){
   const tbody = qs('#cartTable tbody');
   const ids = getCart();
   qs('#cartCountPill').textContent = `${ids.length} item${ids.length!==1?'s':''}`;
   if(!ids.length){ tbody.innerHTML = '<tr><td colspan="4" class="text-center text-secondary">Your cart is empty.</td></tr>'; qs('#cartTotal').textContent = money(0); return; }
```

```
    // Fetch all products once, map by id
    const products = await api(`${BASE_URL}/products`);
    const map = new Map(products.map(p => [p.id, p]));

    let total = 0;
    tbody.innerHTML = ids.map((id, idx) => {
      const p = map.get(id);
      const price = p?.price || 0;
      total += price;
      return `<tr>
        <td>${idx+1}</td>
        <td>${p ? p.name : 'Unknown'} <span class="text-secondary small d-block">${p ? p.description : ''}</span></td>
        <td class="text-end">${money(price)}</td>
        <td class="text-center"><button class="btn btn-sm btn-outline-danger" data-remove-index="${idx}"><i class="bi bi-x"></i></button></td>
      </tr>`;
    }).join('');

    qs('#cartTotal').textContent = money(total);

    // bind remove buttons
    tbody.querySelectorAll('[data-remove-index]').forEach(btn => btn.addEventListener('click', ()=> {
      const i = Number(btn.dataset.removeIndex);
      const items = getCart();
      items.splice(i,1);
      setCart(items);
      renderCart();
    }));
  }
```

```
function placeOrder(){
  const ids = getCart();
  if(!ids.length) { toast('Cart is empty', 'warning'); return; }
  clearCart();
  renderCart();
  toast('Order placed! (demo)', 'success');
}

// ====== EVENT BINDINGS ======
function bindStaticEvents(){
  // welcome
  qs('#gotoLogin').addEventListener('click', ()=> showView('#view-login'));
  qs('#gotoSignup').addEventListener('click', ()=> showView('#view-signup'));

  // signup
  qs('#signupForm').addEventListener('submit', handleSignup);
  qs('#backToLoginFromSignup').addEventListener('click', ()=> showView('#view-login'));

  // login
  qs('#loginForm').addEventListener('submit', handleLogin);
  qs('#backToWelcomeFromLogin').addEventListener('click', ()=> showView('#view-welcome'));

  // admin
  qs('#productForm').addEventListener('submit', handleProductSave);
  qs('#resetProductForm').addEventListener('click', resetProductForm);
  qs('#adminSearch').addEventListener('input', ()=> renderAdmin());

  // user
  qs('#userSearch').addEventListener('input', ()=> renderUserProducts());

  // cart
```

```
      qs('#placeOrderBtn').addEventListener('click', placeOrder);
      qs('#continueShoppingBtn').addEventListener('click', ()=> { renderUserProducts();
showView('#view-user'); });

      // brand
      qs('#brandGoHome').addEventListener('click', ()=> {
        const u = getSession();
        if(!u) showView('#view-welcome');
        else if(u.role==='admin'){ renderAdmin(); showView('#view-admin'); }
        else { renderUserProducts(); showView('#view-user'); }
      });
    }

    // ====== BOOTSTRAP ======
    function boot(){
      setYear();
      bindStaticEvents();
      setNavLinks();
      const u = getSession();
      if(!u){ showView('#view-welcome'); }
      else if(u.role==='admin'){ renderAdmin(); showView('#view-admin'); }
      else { renderUserProducts(); showView('#view-user'); }
    }

    document.addEventListener('DOMContentLoaded', boot);
  </script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```