Lab — Interactivity  
Inventory, Cart, and View Switching

# Goals

* Create an inventory and a shopping cart.
* Add to cart only if stock is available; decrement stock.
* Display and update cart count and total.
* Switch between product view and checkout view.

# Complete example (Vue.js)

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8" />

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

<title>Shop Interactivity — Inventory, Cart, Views</title>

<script src="https://unpkg.com/vue@2.7.8/dist/vue.js"></script>

</head>

<body>

<div id="app">

<button @click="view='products'">Products</button>

<button @click="view='checkout'">Cart ({{ cartCount }})</button>

<section v-if="view==='products'">

<article v-for="p in products" :key="p.id">

<h3>{{ p.name }} — £{{ p.price }}</h3>

<p>In stock: {{ p.stock }}</p>

<button :disabled="p.stock===0" @click="addToCart(p)">Add to cart</button>

</article>

</section>

<section v-else>

<div v-for="(item,i) in cart" :key="item.id">

{{ item.name }} — £{{ item.price }} × {{ item.qty }}

<button @click="changeQty(i,-1)">-</button>

<button @click="changeQty(i,1)" :disabled="productById(item.id).stock===0">+</button>

</div>

<p>Total: £{{ total }}</p>

</section>

</div>

<script>

new Vue({

el: '#app',

data: {

view: 'products',

products: [

{ id: 1, name: 'Dog Toy', price: 9, stock: 5 },

{ id: 2, name: 'Cat Food', price: 12, stock: 3 },

{ id: 3, name: 'Goldfish', price: 4, stock: 0 }

],

cart: []

},

computed: {

cartCount() { return this.cart.reduce((n,i)=>n+i.qty,0); },

total() { return this.cart.reduce((s,i)=>s+i.price\*i.qty,0); }

},

methods: {

productById(id) { return this.products.find(p=>p.id===id); },

addToCart(p) {

if (p.stock <= 0) return;

p.stock--;

const found = this.cart.find(i=>i.id===p.id);

if (found) found.qty++; else this.cart.push({id:p.id,name:p.name,price:p.price,qty:1});

},

changeQty(i, delta) {

const item = this.cart[i];

if (delta === 1) {

const prod = this.productById(item.id);

if (prod.stock <= 0) return;

prod.stock--; item.qty++;

} else {

item.qty--; const prod = this.productById(item.id); prod.stock++;

if (item.qty <= 0) this.cart.splice(i,1);

}

}

}

});

</script>

</body>

</html>

# Assessment checklist

* Add to cart decrements inventory and increments cart line item.
* Add button disabled when stock is zero.
* Cart count and total are derived and always correct.
* Can switch between products and checkout.