Vue Gadget Shop - step9-solution

Ajout d'un compteur pour gérer la quantité des articles dans le panier

Actuellement, lorsqu'on ajoute un gadget au panier, il est ajouté sans gestion de quantité.

Nous allons maintenant:

✓ Ajouter un compteur (quantity) pour chaque produit dans le panier. ✓ Mettre à jour CartDisplay.js pour afficher et modifier la quantité des articles. ✓ Empêcher l'ajout illimité d'un produit si inStock === false.

1. Mise à jour de main.js pour gérer les quantités

Dans main.js, modifions la méthode addToCart pour ajouter ou incrémenter la quantité d'un article.

main.js

```
const app = Vue.createApp({
 data() {
   return {
      title: "Vue Gadget Shop",
      description:
        "Découvrez les derniers gadgets électroniques de haute technologie !",
      gadgets: [
        {
          id: 1,
          name: "Smartphone XZ",
          price: 799,
          image: "./assets/phone.jpg",
          inStock: true,
        },
          id: 2,
          name: "Laptop Pro",
          price: 1299,
          image: "./assets/laptop.jpg",
          inStock: false,
        },
          id: 3,
          name: "Écouteurs Bluetooth",
          price: 199,
          image: "./assets/earbuds.jpg",
          inStock: true,
        },
      ],
      cart: [],
   };
  },
  computed: {
```

```
cartTotal() {
      return this.cart.reduce(
        (total, item) => total + item.price * item.quantity,
     );
   },
   discountedTotal() {
      return this.cartTotal >= 1000 ? this.cartTotal * 0.9 : this.cartTotal;
   },
 },
 methods: {
   addToCart(gadget) {
     let itemInCart = this.cart.find((item) => item.id === gadget.id);
     if (itemInCart) {
        itemInCart.quantity++; // Incrémente la quantité
     } else {
       this.cart.push({ ...gadget, quantity: 1 }); // Ajoute avec quantité = 1
     }
   },
   removeFromCart(index) {
     this.cart.splice(index, 1); // & Supprime l'article complètement du panier
   },
 },
});
```

2. Mise à jour du composant CartDisplay.js

On met à jour CartDisplay.js pour :

- Afficher la quantité de chaque article.
- Permettre d'augmenter/diminuer la quantité.

components/CartDisplay.js

```
<h3 v-if="cart.length > 0">
          (§) Total:
          <span v-if="cartTotal < 1000">{{ cartTotal }}€</span>
          <span v-else>
              <s>{{ cartTotal }}€</s> →
              <strong>{{ discountedTotal }}€ (-10%)</strong>
          </span>
      </h3>
    </div>
 methods: {
    removeItem(index) {
      this.$emit("remove-from-cart", index);
    },
    increaseQuantity(index) {
      this.cart[index].quantity++; // Augmente la quantité
    },
    decreaseQuantity(index) {
      if (this.cart[index].quantity > 1) {
        this.cart[index].quantity--; // Diminue la quantité
      } else {
        this.removeItem(index); // Supprime si quantité = 1
   },
 },
});
```

🖈 Ce qui a changé :

✓ Boutons + et - pour gérer la quantité.
 ✓ Si quantity > 1, on peut diminuer.
 ✓ Si quantity === 1, le bouton - supprime l'article.

🕱 3. Mise à jour de index.html pour utiliser CartDisplay.js

Ajoute à la place de l'ancien code du panier.

index.html

```
<cart-display
  :cart="cart"
  :cart-total="cartTotal"
  :discounted-total="discountedTotal"
  @remove-from-cart="removeFromCart"
>
  </cart-display>
```

4. Mise à jour style.css pour styliser les boutons

Ajoutons un peu de style pour améliorer l'affichage.

style.css

```
/* Style pour les boutons de quantité */
.cart button {
 margin: 0 5px;
 font-size: 1rem;
 padding: 5px 8px;
 border: none;
 cursor: pointer;
 border-radius: 5px;
 transition: background 0.3s;
}
.cart button:hover {
 background: #ddd;
}
/* Bouton "supprimer" */
.remove-btn {
 background: red;
 color: white;
 border: none;
 padding: 5px 10px;
 cursor: pointer;
 border-radius: 5px;
 transition: background 0.3s;
}
.remove-btn:hover {
 background: darkred;
}
```

Résultat attendu