

Vue Gadget Shop - step9-solution

★ Ajout d'un formulaire d'avis sur un produit

Nous allons maintenant ajouter un formulaire permettant aux utilisateurs de laisser un avis sur un gadget.

✦ Objectifs

✓ Créer un formulaire d'avis avec v-model pour capturer les entrées utilisateur. ✓ Associer chaque avis à un produit spécifique. ✓ Utiliser v-for pour afficher les avis en dessous du produit concerné.

✂ 1. Mise à jour de main.js pour stocker les avis

Nous allons ajouter un tableau reviews dans chaque gadget pour stocker les avis soumis.

■ 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,
          reviews: [],
        },
        {
          id: 2,
          name: "Laptop Pro",
          price: 1299,
          image: "./assets/laptop.jpg",
          inStock: false,
          reviews: [],
        },
        {
          id: 3,
          name: "Écouteurs Bluetooth",
          price: 199,
          image: "./assets/earbuds.jpg",
          inStock: true,
          reviews: [],
        },
      ],
      cart: [],
    }
  }
})
```

```

    };
  },
  methods: {
    addToCart(gadget) {
      let itemInCart = this.cart.find((item) => item.id === gadget.id);
      if (itemInCart) {
        itemInCart.quantity++;
      } else {
        this.cart.push({ ...gadget, quantity: 1 });
      }
    },
    removeFromCart(index) {
      this.cart.splice(index, 1);
    },
    addReview(gadget, review) {
      gadget.reviews.push(review); // Ajoute l'avis au produit concerné
    },
  },
});

```

✂ 2. Création du composant ReviewForm.js

Nous allons créer un composant séparé pour gérer le formulaire d'avis. Il enverra les avis au produit concerné grâce à \$emit.

■ components/ReviewForm.js

```

app.component("review-form", {
  props: ["gadget"], // Le gadget sur lequel on laisse un avis
  data() {
    return {
      review: {
        name: "",
        rating: null,
        comment: "",
      },
      submitted: false,
    };
  },
  template: /*html*/ `
    <div class="review-form">
      <h3 class="review-block">Donnez votre avis sur {{ gadget.name }}</h3>
      <form @submit.prevent="submitReview">
        <div class="review-block">
          <label for="name">Nom </label>
          <input type="text" id="name" v-model="review.name" required>
        </div>
        <div class="review-block">
          <label for="rating">Note </label>
          <select id="rating" v-model="review.rating" required>
            <option disabled value="">Sélectionnez une note</option>

```

```

        <option value="5">***** - Excellent</option>
        <option value="4">***** - Très bien</option>
        <option value="3">*** - Moyen</option>
        <option value="2">** - Mauvais</option>
        <option value="1">* - Très mauvais</option>
      </select>
    </div>
    <div class="review-block">
      <label for="comment">Commentaire :</label>
      <textarea id="comment" v-model="review.comment" rows="3" required>
</textarea>
    </div>
    <button type="submit">Envoyer l'avis</button>
  </form>

  <p v-if="submitted" class="confirmation">✔ Merci ! Votre avis a été
enregistré.</p>
</div>
`,
methods: {
  submitReview() {
    this.$emit("add-review", this.review); // ★ Émet un événement pour ajouter
l'avis
    this.submitted = true; // Affiche le message de confirmation
    // Réinitialise le formulaire
    this.review = { name: "", rating: null, comment: "" };
  },
},
});

```

★ Explication

- v-model="review.name" → L'utilisateur entre son nom.
- v-model="review.rating" → Il sélectionne une note sur 5 étoiles.
- v-model="review.comment" → Il écrit son commentaire.
- @submit.prevent="submitReview" → Empêche le rechargement de la page et appelle submitReview().
- this.\$emit("add-review", this.review) → Envoie l'avis au parent (gadget-display).

Attention ! Ne pas oublier d'ajouter ce nouveau composant dans index.html :

```

<script src="./main.js"></script>

<script src="./components/GadgetDisplay.js"></script>
<script src="./components/CartDisplay.js"></script>
<script src="./components/ReviewForm.js"></script>

<!-- Mount App -->
<script>
  const mountedApp = app.mount("#app");
</script>

```

```
</body>
</html>
```

✂ 3. Mise à jour du composant GadgetDisplay.js

Nous allons intégrer le formulaire et afficher les avis sous chaque gadget.

■ components/GadgetDisplay.js

```
app.component("gadget-display", {
  props: ["gadget"],
  template: /*html*/ `
    <div class="gadget">
      
      <h2>{{ gadget.name }}</h2>
      <p>Prix : {{ gadget.price }}€</p>
      <p :class="{ 'in-stock': gadget.inStock, 'out-of-stock': !gadget.inStock }">
        {{ gadget.inStock ? '✓ En stock' : '✗ En rupture de stock' }}
      </p>
      <button @click="addToCart" :disabled="!gadget.inStock">Ajouter au
panier</button>

      <!-- Affichage des avis -->
      <h3 v-if="gadget.reviews.length > 0">Avis :</h3>
      <ul v-if="gadget.reviews.length > 0">
        <li v-for="review in gadget.reviews" :key="review.name">
          <strong>{{ review.name }}</strong> - {{ review.rating }}★
          <p>{{ review.comment }}</p>
        </li>
      </ul>
      <p v-else>Aucun avis pour ce produit.</p>

      <!-- Formulaire d'avis -->
      <review-form :gadget="gadget" @add-review="addReview"></review-form>
    </div>
  `,
  methods: {
    addToCart() {
      this.$emit("add-to-cart", this.gadget);
    },
    addReview(review) {
      this.gadget.reviews.push(review); // ✎ Ajoute l'avis au produit
    },
  },
});
```

✂ 4. Ajout de styles pour le formulaire et les avis

Ajoutons des styles dans style.css pour rendre l'affichage plus agréable.

■ style.css

```
/* Conteneur du formulaire d'avis */
.review-form {
  background: #f9f9f9;
  padding: 15px;
  border-radius: 8px;
  margin-top: 10px;
}

.review-block {
  margin-bottom: 10px;
}

/* Liste des avis */
.gadget ul {
  list-style: none;
  padding: 0;
}

.gadget li {
  background: white;
  padding: 10px;
  margin-top: 5px;
  border-radius: 5px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

/* Message de confirmation */
.confirmation {
  margin-top: 10px;
  color: green;
  font-weight: bold;
}
```

✓ Résultat attendu

- Un formulaire d'avis sous chaque produit.
- L'utilisateur peut noter et commenter un gadget.
- Les avis sont enregistrés et affichés sous le produit.