document.addEventListener('DOMContentLoaded', () => {

const updateTotalPrice = () => {

let total = 0;

document.querySelectorAll('.card').forEach(card => {

const unitPrice = parseFloat(card.querySelector('.unit-price').textContent.replace('$', ''));

const quantity = parseInt(card.querySelector('.quantity').textContent);

total += unitPrice \* quantity;

});

document.querySelector('.total').textContent = `${total} $`;

};

document.querySelectorAll('.fa-plus-circle').forEach(button => {

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

const quantityElement = button.nextElementSibling;

quantityElement.textContent = parseInt(quantityElement.textContent) + 1;

updateTotalPrice();

});

});

document.querySelectorAll('.fa-minus-circle').forEach(button => {

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

const quantityElement = button.previousElementSibling;

if (parseInt(quantityElement.textContent) > 0) {

quantityElement.textContent = parseInt(quantityElement.textContent) - 1;

updateTotalPrice();

}

});

});

document.querySelectorAll('.fa-trash-alt').forEach(button => {

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

const card = button.closest('.card-body');

card.remove();

updateTotalPrice();

});

});

document.querySelectorAll('.fa-heart').forEach(button => {

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

button.style.color = 'red';

setTimeout(() => {

button.style.color = '';

}, 300000); // 5 minutes in milliseconds

});

});

updateTotalPrice();

});

/\*document.addEventListener('DOMContentLoaded', () => {

const updateTotalPrice = () => {

let total = 0;

document.querySelectorAll('.card').forEach(card => {

const unitPrice = parseFloat(card.querySelector('.unit-price').textContent.replace('$', ''));

const quantity = parseInt(card.querySelector('.quantity').textContent);

total += unitPrice \* quantity;

});

document.querySelector('.total').textContent = `${total} $`;

};

const updateQuantity = (element, delta) => {

const quantityElement = element.closest('.card-body').querySelector('.quantity');

let quantity = parseInt(quantityElement.textContent);

quantity = Math.max(0, quantity + delta);

quantityElement.textContent = quantity;

updateTotalPrice();

};

const toggleHeart = (heartElement) => {

heartElement.classList.toggle('fas');

heartElement.classList.toggle('far');

};

const deleteCard = (deleteElement) => {

const card = deleteElement.closest('.card');

card.remove();

updateTotalPrice();

};

document.querySelectorAll('.fa-plus-circle').forEach(button => {

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

updateQuantity(button, 1);

});

});

document.querySelectorAll('.fa-minus-circle').forEach(button => {

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

updateQuantity(button, -1);

});

});

document.querySelectorAll('.fa-heart').forEach(heart => {

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

toggleHeart(heart);

});

});

document.querySelectorAll('.fa-trash-alt').forEach(trash => {

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

deleteCard(trash);

});

});

updateTotalPrice(); // Initial call to set the total price

});\*/ je voudais l'explication detaillé de ce code

**Copilot said:**

Copilot

Bien sûr! Voici une explication détaillée de votre code JavaScript.

**Explication du Code**

**Initialisation**

document.addEventListener('DOMContentLoaded', () => {

Cette ligne attend que le contenu du DOM soit entièrement chargé avant d'exécuter le code à l'intérieur de la fonction.

**Mise à jour du prix total**

const updateTotalPrice = () => {

let total = 0;

document.querySelectorAll('.card').forEach(card => {

const unitPrice = parseFloat(card.querySelector('.unit-price').textContent.replace('$', ''));

const quantity = parseInt(card.querySelector('.quantity').textContent);

total += unitPrice \* quantity;

});

document.querySelector('.total').textContent = `${total} $`;

};

Cette fonction calcule le prix total en parcourant chaque élément avec la classe .card, en récupérant le prix unitaire et la quantité, puis en mettant à jour le total.

**Gestion des boutons d'ajout**

document.querySelectorAll('.fa-plus-circle').forEach(button => {

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

const quantityElement = button.nextElementSibling;

quantityElement.textContent = parseInt(quantityElement.textContent) + 1;

updateTotalPrice();

});

});

Cette partie ajoute un écouteur d'événements click à chaque bouton avec la classe .fa-plus-circle. Lorsqu'on clique, la quantité est augmentée de 1 et le prix total est mis à jour.

**Gestion des boutons de soustraction**

document.querySelectorAll('.fa-minus-circle').forEach(button => {

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

const quantityElement = button.previousElementSibling;

if (parseInt(quantityElement.textContent) > 0) {

quantityElement.textContent = parseInt(quantityElement.textContent) - 1;

updateTotalPrice();

}

});

});

Cette partie ajoute un écouteur d'événements click à chaque bouton avec la classe .fa-minus-circle. Lorsqu'on clique, la quantité est diminuée de 1 (si elle est supérieure à 0) et le prix total est mis à jour.

**Gestion des boutons de suppression**

document.querySelectorAll('.fa-trash-alt').forEach(button => {

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

const card = button.closest('.card-body');

card.remove();

updateTotalPrice();

});

});

Cette partie ajoute un écouteur d'événements click à chaque bouton avec la classe .fa-trash-alt. Lorsqu'on clique, l'élément .card-body parent est supprimé et le prix total est mis à jour.

**Gestion des boutons de cœur**

document.querySelectorAll('.fa-heart').forEach(button => {

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

button.style.color = 'red';

setTimeout(() => {

button.style.color = '';

}, 300000); // 5 minutes in milliseconds

});

});

Cette partie ajoute un écouteur d'événements click à chaque bouton avec la classe .fa-heart. Lorsqu'on clique, la couleur du bouton devient rouge pendant 5 minutes (300000 millisecondes).

**Mise à jour initiale du prix total**

updateTotalPrice();

Cette ligne appelle la fonction updateTotalPrice pour initialiser le prix total dès que le DOM est chargé.

**Code Commenté**

Le code commenté à la fin est une version alternative du code principal avec quelques différences dans la gestion des événements et des fonctions. Il utilise des fonctions pour mettre à jour les quantités, basculer les cœurs et supprimer les cartes de manière légèrement différente.

N'hésitez pas à me demander si vous avez besoin de plus de détails ou d'explications supplémentaires! 😊

Modifier dans Pages

Copier

Il est possible que le contenu généré par IA soit incorrect

Can you optimize this code further?How to add a discount feature?What are best practices for JavaScript?