

Angèle Mounier B1

La fleur du bonheur

Shopping List



B1 Angèle Mounier

La Fleur du Bonheur

Sommaire



Objectif

Mon projet

Identité visuelle de mon projet

Présentation de ma Shopping List

Explication de mon code

Objectif

Réaliser une Shopping List

Fonctionnalités indispensables:

- saisir un nouveau produit dans la liste
- ôter un produit précédemment rentré dans la liste

Des fonctionnalités additionnelles peuvent être implémentées :

- saisir le prix du produit pour prévoir le montant total des achats
- saisir des quantités pour chaque produit
- catégoriser les achats (épicerie, hygiène, produits ménagers...)
- toute fonctionnalité pertinente selon votre imagination...





Mon projet

J'ai choisi d'incarner une boutique fictive: La Fleur du bonheur

Ce commerce proposerait des produits faits en fleurs séchées pour les mariages comme des boucles d'oreilles, des coiffures ou encore des décorations.

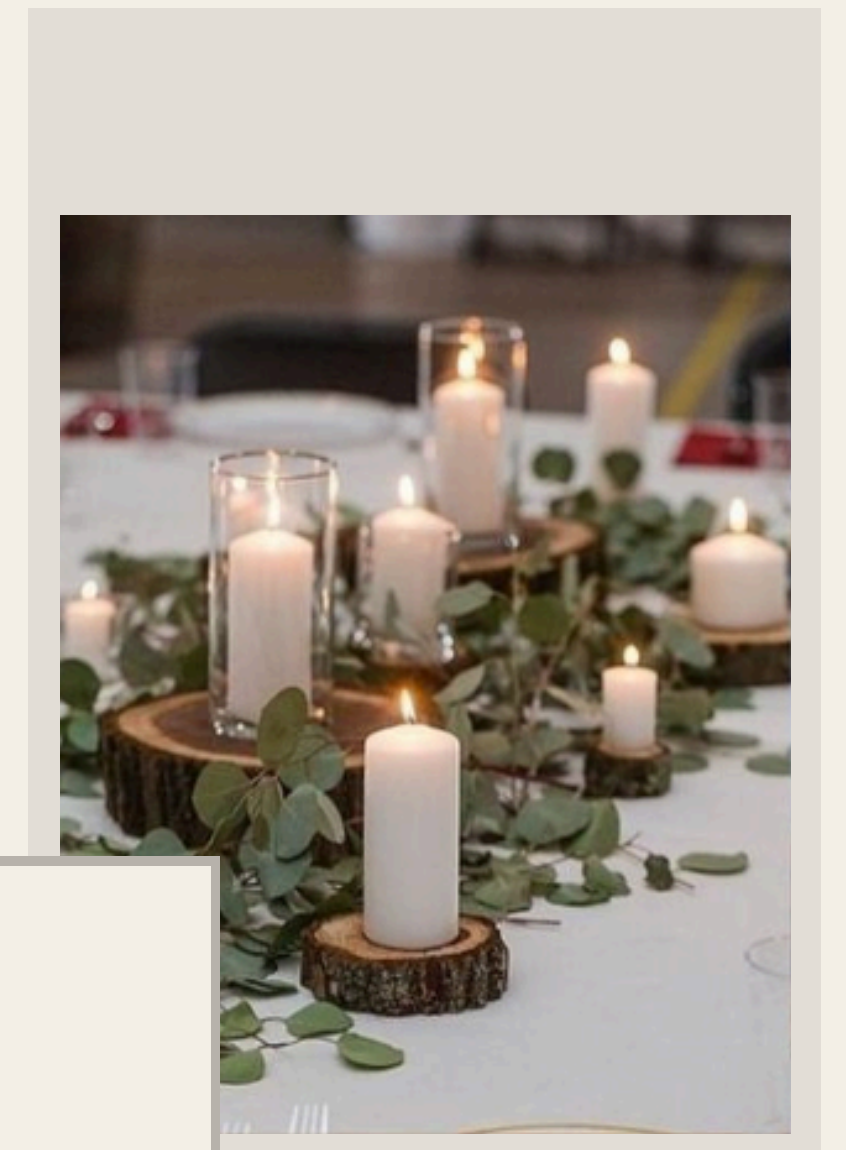
Sur son site web, La Fleur du Bonheur mettrait à disposition de ses clients une shopping list.

BI Angèle Mounier

La Fleur du Bonheur

Identité visuelle

Planche d'ambiance



Identité visuelle

Le logo



Style de la marque

Classique : nom de produits inspirés de la littérature et de l'Antiquité

Style des produits

Bohême

Identité visuelle

Couleurs



Polices

Great vibes



Titre

Montserrat



Sous-titre

Texte

Présentation de ma Shopping List



La fleur du bonheur

Votre liste de produits fleuris!

Rechercher un produit...

Toutes les catégories ▾



**Boucles d'oreilles -
Thalie Blanche**

33.90 €
▶ Voir description

Ajouter au panier



**Décoration - Rêverie
romantique**

15.90 €
▶ Voir description

Ajouter au panier



**Coiffure - Barette
Juliette**

18.90 €
▶ Voir description

Ajouter au panier



**Boucles d'oreilles -
Aglée Blanche**

35.00 €
▶ Voir description

Ajouter au panier



**Boucles d'oreilles -
Aphrodite**

32.90 €
▶ Voir description

Ajouter au panier

B1 Angèle Mounier

Ma Shopping List présente une barre de recherche pour trouver le produit que l'on veut acheter, elle propose également des filtres pour mieux retrouver le produit recherché.

Les produits apparaissent sous formes de carte, comportant une photo du produit, son nom, son prix, une brève description et un bouton permettant de l'ajouter à la liste.

Il est possible de voir le total de la liste de course en temps réel.

La Fleur du Bonheur



Explication de mon code

Structure de la page

Organisation des données

Définition de variables globales

Ecouteurs d'évènements

Fonction renderProducts

Fonction renderCart

Structure de la page

Ma page comporte quatre grandes zones:

- l'entête avec titre, sous-titre et le logo
- la partie recherche avec une barre de recherche et des filtres
- la partie produits avec une grille présentant les produits sous formes de cartes
- la partie Shopping List, où s'affiche les produits sélectionnés ainsi que des boutons permettant de régler la quantité de produits, et enfin, le total des produits de la liste.



Organisation des données

Les données des cartes des produits sont répertoriées dans le fichier JS, sous forme d'objets, ce qui permet ensuite de changer leur affichage dynamiquement. Pour chaque produit, on répertorie l'id, le titre, le prix, une description, le lien vers l'image, et, enfin, la catégorie à laquelle il appartient, ce qui permettra de filtrer les produits. Ici, par exemple, on voit la catégorie : "boucles", ce qui signifie que, lorsqu'on sélectionnera toutes les boucles d'oreilles, ce produit apparaîtra.

"Const" permet de déclarer le tableau "products", qui contiendra les objets.

```
const products = [  
  {  
    id: 1,  
    name: "Boucles d'oreilles - Thalie Blanche",  
    price: 33.90,  
    description: "Boucles d'oreilles en fleurs stabilis  
    image: "img/boucles/boucle (1).png",  
    category: "boucles"  
  },  
]
```



Boucles d'oreilles - Aphrodite

32.90 €

► Voir description

Ajouter au panier

Définition de variables globales

- cart : objet où on mettra les produits qui apparaîtront dans le panier
- productContainer : variable qui affichera toutes les cartes produit.
- cartList : liste des produits ajoutés au panier.
- totalPriceElement : affichage du prix total.
- searchInput : définition du champ de recherche
- categoryFilter : menu déroulant permettant de filtrer par catégorie.



```
let cart = {};  
  
let productContainer = document.getElementById("productContainer");  
let cartList = document.getElementById("cartList");  
let totalPriceElement = document.getElementById("totalPrice");  
let searchInput = document.getElementById("searchInput");  
let categoryFilter = document.getElementById("categoryFilter");
```

Écouteurs d'événement

Avant de définir les fonctions, on pose des "écouteurs d'évènements" : `addEventListener`, qui, par l'input ou le `change`, appelleront la fonction `renderProducts`, permettant d'afficher les cartes produits.

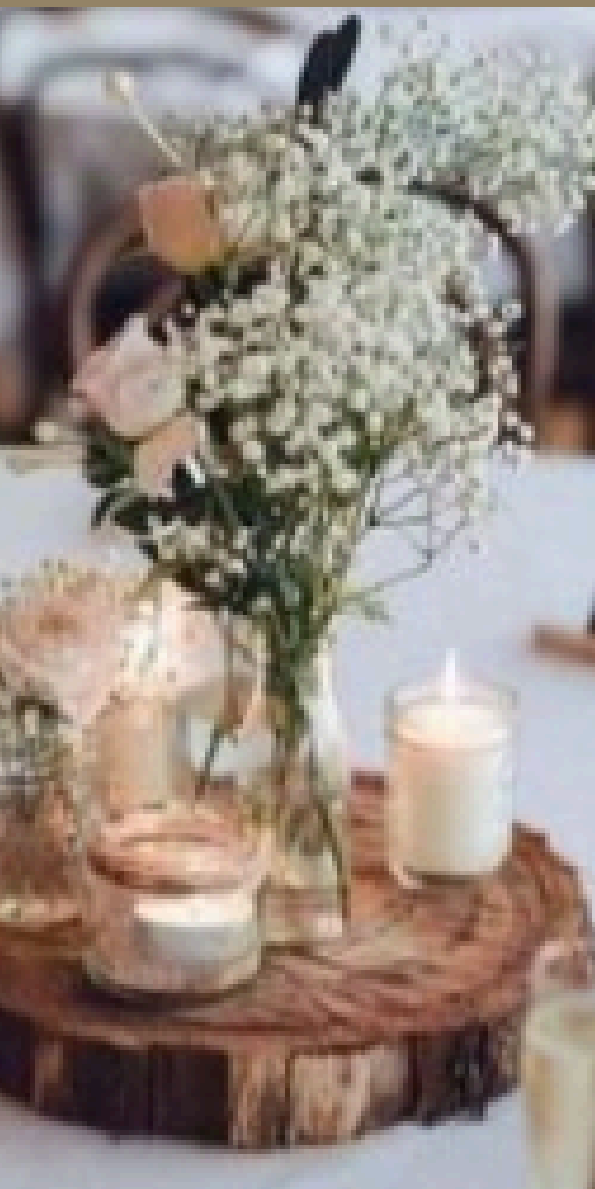
Ainsi, chaque fois que l'utilisateur tape dans la barre de recherche (input) ou change la catégorie sélectionnée (change), la fonction `renderProducts` est relancée pour afficher les produits souhaités

```
searchInput.addEventListener("input", renderProducts);  
categoryFilter.addEventListener("change", renderProducts);
```

Fonction appelée

Écouteurs d'évènement
permettant d'appeler
la fonction `renderProducts`

Évènement "écouté"

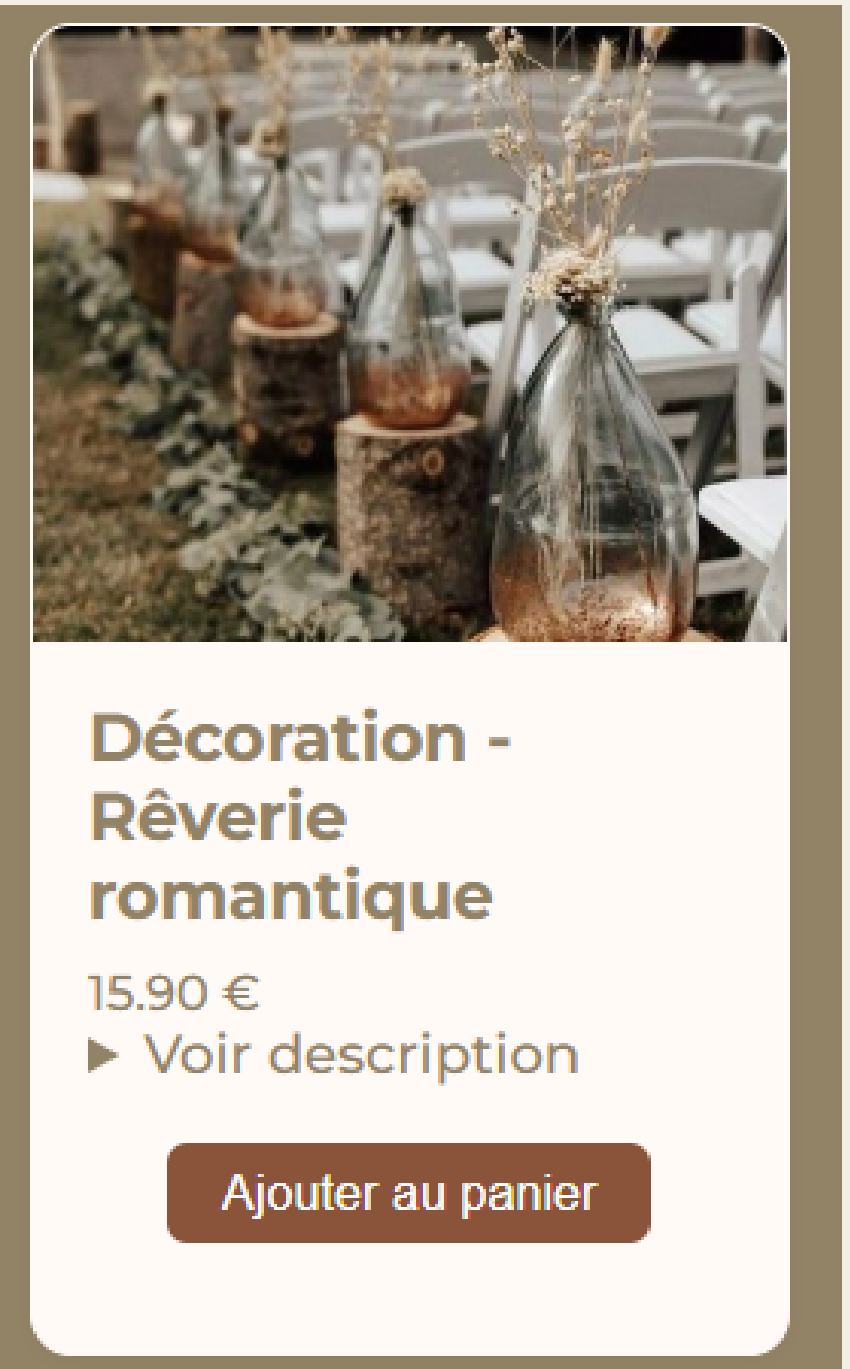


Fonction renderProducts (1)

Cette fonction a pour but principal d'afficher, sous forme de carte, les informations déclarées précédemment, dans le tableau "Products".

Elle vide le conteneur des produits, récupère les filtres actifs et filtre les produits selon la recherche et la catégorie demandée.

```
function renderProducts() {  
  productContainer.innerHTML = '';  
  
  let searchTerm = searchInput.value.toLowerCase();  
  let selectedCategory = categoryFilter.value;  
  
  let filteredProducts = products.filter(product => {  
    let matchCategory = selectedCategory === "all" || product.category === selectedCategory;  
    let matchSearch = product.name.toLowerCase().includes(searchTerm);  
    return matchCategory && matchSearch;  
  });  
}
```



```

filteredProducts.forEach(product => {
  let card = document.createElement("div");
  card.className = "product-card";

  let img = document.createElement("img");
  img.src = product.image;
  card.appendChild(img);

  let title = document.createElement("h3");
  title.textContent = product.name;

  let price = document.createElement("p");
  price.textContent = product.price.toFixed(2) + " €";

  let descriptionToggle = document.createElement("details");
  let summary = document.createElement("summary");
  summary.textContent = "Voir description";
  let desc = document.createElement("p");
  desc.textContent = product.description;
  descriptionToggle.appendChild(summary);
  descriptionToggle.appendChild(desc);

```



Fonction renderProducts (2)

Ensuite, elle crée dynamiquement les cartes produits avec, pour chaque produit filtré :

- une div avec le classe product-card (nécessaire pour appliquer le CSS)
- l'image, le titre, le prix
- un details/summary pour afficher la description
- un bouton "Ajouter au panier"

Le tout est rassemblé dans une variable info pour faciliter la mise en page CSS

```

let addButton = document.createElement("button");
addButton.textContent = "Ajouter au panier";
addButton.addEventListener("click", function () {
  if (!cart[product.id]) {
    cart[product.id] = 1;
  } else {
    cart[product.id]++;
  }
  renderCart();
});

```

```

let info = document.createElement("div");
info.className = "info";
info.appendChild(title);
info.appendChild(price);
info.appendChild(descriptionToggle);
info.appendChild(addButton);

card.appendChild(info);

```



D'où vient cette fonction?

Fonction renderCart (1)

Il s'agit de la fonction remarquée précédemment, permettant d'afficher les produits dans la liste.

Celle-ci vide le panier affiché. Puis, grâce à une boucle for, elle parcourt chaque élément du panier et, pour chaque produit (via son id), elle récupère l'objet "product" correspondant, calcule le prix total du produit.

```
for (let id in cart) {  
  let product = products.find(p => p.id == id);  
  let quantity = cart[id];  
  let itemTotal = product.price * quantity;  
  total += itemTotal;  
}
```

Votre Liste

Boucles d'oreilles - Athéna

Total : 29.00 €





Fonction renderCart (2)

Elle crée ensuite une li contenant le nom du produit, un contrôleur de quantité avec les boutons + et -, et le prix total de cet article. Elle ajoute ensuite les éléments dans quantityDiv

```
let li = document.createElement("li");

let nameSpan = document.createElement("span");
nameSpan.textContent = product.name;

let quantityDiv = document.createElement("div");
quantityDiv.className = "quantity-control";
```

Bouton -

```
let minusButton = document.createElement("button");
minusButton.textContent = "-";
minusButton.addEventListener("click", function () {
  cart[id]--;
  if (cart[id] <= 0) {
    delete cart[id];
  }
  renderCart();
});
```

Bouton +

```
let qtySpan = document.createElement("span");
qtySpan.textContent = quantity;

let plusButton = document.createElement("button");
plusButton.textContent = "+";
plusButton.addEventListener("click", function () {
  cart[id]++;
  renderCart();
});
```

Fonction renderCart (3)

La fonction assemble les éléments dans la li, ajoute l'élément à la liste HTML de la ShoppingList et affiche le prix total de tous les articles

```
quantityDiv.appendChild(minusButton);
quantityDiv.appendChild(qtySpan);
quantityDiv.appendChild(plusButton);

let priceSpan = document.createElement("span");
priceSpan.textContent = itemTotal.toFixed(2) + " €";

li.appendChild(nameSpan);
li.appendChild(quantityDiv);
li.appendChild(priceSpan);

cartList.appendChild(li);
}

totalPriceElement.textContent = total.toFixed(2) + " €";
}

renderProducts();
```





Merci de votre attention