



Ludorama

Votre monde de jeu, sans frontière

Livable Semaine 50 PHP

Pierre CORNU | Raphaël LAMOTHE | Naria SAVARY | David TRAN
Groupe 1A

TABLE DES MATIÈRES

Introduction.....	3
Fonctionnalités.....	4
Catégories de produits.....	4
Explication du code.....	5
Détails du produit.....	5
Voir annexe 1.....	5
Explication du code.....	6
Panier.....	6
Explication du code.....	6
ANNEXE.....	7
Annexe 1.....	7
Annexe 2.....	9



INTRODUCTION

Cette semaine, nous avons concentré nos efforts sur le développement d'un ensemble de fonctionnalités que nous allons vous détailler, répondant aux besoins exprimés dans les User Story suivantes :

"En tant que client, je veux pouvoir ajouter des produits à mon panier et voir le total en temps réel, afin de mieux gérer mon budget."

"En tant que client, je veux voir des photos et des descriptions détaillées des confiseries, afin de prendre une décision éclairée avant d'acheter."

"En tant que client, je veux pouvoir rechercher des confiseries par nom ou catégorie, afin de trouver rapidement ce que je souhaite acheter."

Pour répondre à ces besoins, nous avons développé de nombreuses fonctionnalités sur lesquelles nous devons continuer à travailler afin de parfaire notre site :

1. **Ajout de produit(s) au panier / Achat de son panier**

Nous avons implémenté un panier afin de permettre à l'utilisateur de stocker les articles qu'il souhaite et qu'il puisse donc acheter les articles qu'il a consultés et choisis. Nous travaillons encore sur la possibilité de payer le panier et de valider la commande de l'utilisateur.

2. **Descriptions détaillées des confiseries**

A l'aide des données stockées dans notre base de données, nous affichons les données particulières à chaque produit afin que l'utilisateur puisse connaître précisément les produits qu'il pourrait commander.

3. **Recherche des confiseries par catégorie**

Nous avons dès à présent créé une première catégorie de recherche (la catégorie d'appartenance d'un bonbon) afin que l'utilisateur puisse chercher les bonbons qu'il préfère. Nous travaillons en ce moment sur le développement d'une catégorie de prix, afin que les utilisateurs puissent penser à leur argent en achetant les produits SweetShops.

Le travail effectué lors de cette semaine de Sprint constitue une base très importante pour le bon fonctionnement de notre site web. L'utilisateur a la possibilité d'avoir un panier fonctionnel qu'il pourra très prochainement commander et il peut rechercher ses produits selon les catégories des bonbons. De plus, les produits possèdent un affichage détaillé avec l'ensemble de ses données propres.



FONCTIONNALITÉS

Catégories de produits

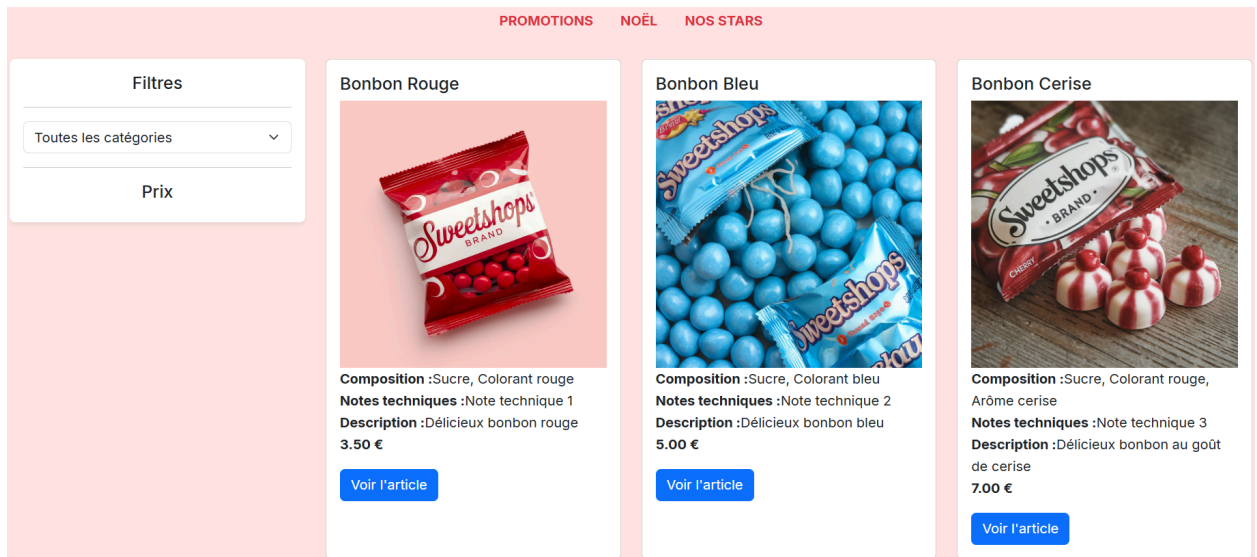


Figure 1 : Affichage global des produits

```
<div class="row">
  <!-- Barre de filtres -->
  <div class="col-12 col-md-3 mb-4">
    <div class="bg-white p-3 rounded shadow-sm">
      <h5 class="text-center">Filtres</h5>
      <hr>
      <form method="POST">
        <select name="categorie" class="form-select mb-3" onchange="this.form.submit()">
          <option value="0" <?= $selectedCategory === 0 ? 'selected' : '' ?>>Toutes les catégories</option>
          <?php foreach ($categories as $category): ?>
            <option value="<?= $category['IDCATEG'] ?>" <?= $selectedCategory === $category['IDCATEG'] ? 'selected' : '' ?>>
              <?= htmlspecialchars($category['NOMCATEG']) ?>
            </option>
          <?php endforeach; ?>
        </select>
      </form>
      <hr>
      <h5 class="text-center">Prix</h5>
    </div>
  </div>
</div>
```


Figure 1 Code : Traitement des catégories



Explication du code

Le code présenté pour le traitement des catégories se présente en un bloc de code assez compact. Cependant, celui-ci s'explique simplement : nous créons un formulaire à l'aide de la balise `<form>` et, à l'aide de l'attribut `"onchange"`, on actualise le formulaire à chaque changement de valeur. On ajoute ensuite à notre formulaire soit la catégorie "Toutes les catégories", soit une case vide. On ajoute ensuite l'ensemble des catégories existantes dans notre formulaire. Lorsque la valeur du formulaire s'exécute, on actualise l'affichage qui suit dans la page, et donc, les bonbons à afficher.

Détails du produit



Bonbon Rouge

★★★★☆

Notes Techniques : Note technique 1

Description : Délicieux bonbon rouge

Ingédients : Sucre, Colorant rouge

Formats Disponibles

Petit

Couleurs Disponibles

Rouge Bleu

En Stock

[Ajouter au panier](#) [Acheter](#)

Figure 2 : Affichage détaillé des produits

Figure 2 Code : Affichage détaillé des produits







Voir annexe 1.

Explication du code

Le code présenté pour l’affichage détaillé des produits possède différentes parties. D’abord, on peut observer l’affichage de l’image de notre produit. On récupère ensuite les informations de notre produit afin de pouvoir afficher l’ensemble. On affiche son nom, puis la moyenne des notes qui lui sont attribuées, que l’on fait apparaître sous forme d’étoiles à l’aide d’une division par 2. On affiche ensuite l’ensemble des données de l’article selon ses disponibilités. Enfin, on met en place deux boutons permettant ou d’ajouter au panier, ou bien de directement aller payer l’ensemble des articles du panier. On ajoute à cela un champ pour entrer le nombre de produits souhaités

Panier

ARTICLES DE MON PANIER :

	Bonbon Rouge <input style="width: 100px;" type="text" value="1"/> 3.50€	
	Bonbon Veggie <input style="width: 100px;" type="text" value="1"/> 9.00€	
	Bonbon Vanille <input style="width: 100px;" type="text" value="1"/> 16.00€	

RÉSUMÉ DE LA COMMANDE

Total des articles (Prix initial)	16.00 €
Frais de livraison	2.5 €
Total de la commande	18.5 €

Encore une envie de nostalgie ?

Valider mon panier

Figure 3 : Affichage du panier

Figure 3 Code : Affichage du panier

Voir annexe 2.

Explication du code

Le code présenté pour la réalisation du panier est pour sa part assez complexe. Il appelle un premier script JavaScript afin d’empêcher une saisie négative à l’utilisateur, et un deuxième afin d’actualiser les données choisies lorsque l’on actualise la page. On affiche à l’image représentant le produit, ainsi que son nom, et un champ de nombres appelant les scripts JavaScript. On affiche un total



(mal calculé à l'heure actuelle), et on permet à l'utilisateur de valider ses options. La suite de la page est encore en cours de travail.

ANNEXE

Annexe 1

```
<div class="col-md-6 text-center product-image" id="product-img">
    <div class="cadre-image">
        <?php $idProduit = (int)$_GET['idProduit']; echo "<img src='images/produits/image" .
htmlspecialchars($idProduit) . ".jpeg' width='100%'>" ?>
    </div>
</div>
```

```
<?php
if (isset($_GET['idProduit']) && is_numeric($_GET['idProduit'])) {
    try {
        // Requête produit
        $stmt = $conn->prepare("SELECT * FROM PRODUIT WHERE IDPROD = ?");
        $stmt->execute([$idProduit]);
        $produit = $stmt->fetch(PDO::FETCH_ASSOC);

        if (!$produit) {
            echo "<p class='text-danger'>Aucun produit trouvé pour cet ID.</p>";
            exit();
        }

        // Affichage des données produit
        echo "<h2 class='textProduit'>" . htmlspecialchars($produit['NOMPROD']) . "</h2>";

        // Requête moyenne des avis
        $stmt = $conn->prepare('CALL get_moyenne_prod(?)');
        $stmt->execute([$idProduit]);
        $moyenneNotes = $stmt->fetch(PDO::FETCH_ASSOC);

        // Affichage des étoiles
        echo "<div class='star-rating'>";
        if ($moyenneNotes !== false) {
            $moyenneNotes = round($moyenneNotes['noteAvg'], 0, PHP_ROUND_HALF_DOWN);
```



```
        for ($i = 0; $i < $moyenneNotes - 1; $i += 2) {
            echo "<img src='images/etoile_pleine.png' width='20' height='20'></img>";
        }
        if ($moyenneNotes - $i == 1) {
            echo "<img src='images/demi_etoile.png' width='20' height='20'></img>";
            $i += 2;
        }
        for ($i; $i < 10; $i += 2) {
            echo "<img src='images/etoile_vide.png' width='20' height='20'></img>";
        }
    } else {
        echo "Aucun avis sur ce produit";
    }
}
echo "</div>";

echo "<p><strong>Notes Techniques : </strong>" . htmlspecialchars($produit["NOTESTECH"]) . "</p>";
echo "<p><strong>Description : </strong>" . htmlspecialchars($produit["DESCRIPTION"]) . "</p>";
echo "<p><strong>Ingrédients : </strong>" . htmlspecialchars($produit["COMPOSITION"]) . "</p>";

// Requête caractéristiques (formats et couleurs)
$stmt = $conn->prepare('CALL get_dispos_produit(?)');
$stmt->execute([$idProduit]);
$carac = $stmt->fetchAll(PDO::FETCH_ASSOC);

echo "<h3>Formats Disponibles</h3>";
$formats = [];
foreach ($carac as $car) {
    if (!in_array($car["NOMFORMAT"], $formats)) {
        echo "<button class='btn btn-format'>" . htmlspecialchars($car["NOMFORMAT"]) . "</button>";
        $formats[] = $car["NOMFORMAT"];
    }
}

echo "<h3>Couleurs Disponibles</h3>";
$couleurs = [];
foreach ($carac as $car) {
    if (!in_array($car["NOMCOULEUR"], $couleurs)) {
        echo "<button class='btn btn-color'>" . htmlspecialchars($car["NOMCOULEUR"]) . "</button>";
        $couleurs[] = $car["NOMCOULEUR"];
    }
}

// Formulaire pour ajout au panier
echo "<div class='stock-info'>
    <h3>En Stock</h3>
```




```
        <form action='includes/TraitAjoutPanier.php?idProduit=' . htmlspecialchars($idProduit)
. "" method='POST' class='form-inline'>
            <input type='number' id='quantite' name='quantite' class='form-control' min='1'
step='1' value='1' required onblur='checkNegativeOnBlur(this)'>
            <button type='submit' name='action' value='Ajouter au panier' class='btn btn-success
ms-2'>Ajouter au panier</button>
            <button type='submit' name='action' value='Acheter' class='btn btn-danger
ms-2'>Acheter</button>
        </form>
    </div>";
    } catch (PDOException $e) {
        echo "<p class='text-danger'>Erreur : " . htmlspecialchars($e->getMessage()) . "</p>";
    }
    } else {
        echo "<p class='text-danger'>ID produit manquant ou invalide.</p>";
    }
}
```

Annexe 2

```
<div class="panier">
    <div class="container">
        <div class="row py-5 align-items-center">
            <!-- Logo -->
            <div class="col-lg-2 col-md-3 col-auto text-center">
                <a href="index.php">
                    
                </a>
            </div>
            <!-- Résumé -->
            <div class="col-lg-10 col-md-9 col text-start">
                <h3 class="text-uppercase text-pink">Résumé de votre commande</h3>
            </div>
        </div>

        <!-- Progressbar -->
        <div class="container">
            <div class="progressbar">
                <!-- Step 1 -->
                <div class="progress-step active">
                    <div class="step-circle">1</div>
                    <div class="step-label">Panier</div>
                </div>
                <!-- Step 2 -->
            </div>
        </div>
    </div>
</div>
```



```
<div class="progress-step">
  <div class="step-circle">2</div>
  <div class="step-label">Livraison</div>
</div>
<!-- Step 3 -->
<div class="progress-step">
  <div class="step-circle">3</div>
  <div class="step-label">Paiement</div>
</div>
</div>

<hr>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $prixUnitaire = 6.95;
    $fraisLivraison = 2.50;
    $quantity = isset($_POST['quantity']) ? (int)$_POST['quantity'] : 1;
    if ($quantity < 1) {
        $quantity = 1;
    }
    $totalArticles = $prixUnitaire * $quantity;
    $totalCommande = $totalArticles + $fraisLivraison;
}
?>
<!-- Contenu de la commande -->
<div id="step1-content">
  <div class="row">
    <div class="col-lg-8">
      <h4 class="text-uppercase text-pink mb-4">Articles de mon panier :</h4>
      <form id="cart-form">
        <?php
          require_once('connect.inc.php');

          $stmt = $conn->prepare('CALL get_panier(?)');
          $stmt->execute([$_SESSION['idCompte']]);
          $panier = $stmt->fetch(PDO::FETCH_ASSOC);

          if ($panier){
            $stmt = $conn->prepare('CALL get_commande_details(?)');
            $stmt->execute([$panier['IDCOMMANDE']]);

            $produits = $stmt->fetch(PDO::FETCH_ASSOC);

            foreach ($produits as $produit){
```



```
$stmt = $conn->prepare("CALL get_dispos_produit_light(?)");
$stmt->execute([$produit['IDPROD']]);
$infos = $stmt->fetch(PDO::FETCH_ASSOC);

$prix = $infos['PRIX'];

echo "<div class='d-flex align-items-center py-3 border-bottom'>
<img src='./images/test.jpg' alt='Produit' class='img-fluid' style='max-width: 120px;'>
  <div class='ms-3 flex-grow-1'>
    <p class='mb-1'>";

$stmt = $conn->prepare("SELECT NOMPROD FROM PRODUIT WHERE IDPROD = ?");
$stmt->execute([$produit['IDPROD']]);
$nom = $stmt->fetch(PDO::FETCH_ASSOC);

echo $nom['NOMPROD'];
echo "</p>
  <div class='d-flex align-items-center'>
    <form action='shoppingCart.php' method='post'>
      <input type='number' id='quantity' name='quantity' min='1' step='1'
value='".$produit['QTE']."' required onblur='checkNegativeOnBlur(this)' oninput='sendInputValue(this)'>
    </form>
    <button class='btn btn-link text-danger'>
      <svg xmlns='http://www.w3.org/2000/svg' width='16' height='16'
fill='currentColor' class='bi bi-trash' viewBox='0 0 16 16'>
        <path d='M5.5 5.5A5.5 0 0 1 6 6v6a5.5 0 0 1 1 0V6a5.5 0 0 1 .5-.5m2.5
0a5.5 0 0 1 .5.5v6a5.5 0 0 1 1 0V6a5.5 0 0 1 .5-.5m3 .5a5.5 0 0 1 0v6a5.5 0 0 1 0z' />
        <path d='M14.5 3a1 0 0 1 1 1H13v9a2 2 0 0 1 2 2H5a2 2 0 0
1 2-2V4h-.5a1 0 0 1-1-1V2a1 0 0 1 1-1H6a1 0 0 1 1-1h2a1 0 0 1 1 1h3.5a1 0 0 1 1 1zM4.118 4 4
4.059V13a1 0 0 0 1 1h6a1 0 0 0 1-1V4.059L11.882 4zM2.5 3h11V2h-11z' />
      </svg>
    </button>
  </div>
</div>
  <span class='text-pink fw-bold'>".$prix."€ </span>
</div>";
}
}
?>
</form>
</div>
<div class="col-lg-4">
  <h5 class="text-uppercase text-pink mb-3">Résumé de la commande</h5>
  <table class="table text-end">
    <tr>
      <td>Total des articles (Prix initial)</td>
```



```
<td><?php $total_articles = isset($quantity) ? $prix * $quantity : $prix; echo $total_articles ?> €</td>
</tr>
<tr>
    <td>Frais de livraison</td>
    <td><?php $frais_livraison = 2.50; echo $frais_livraison ?> €</td>
</tr>
<tr class="fw-bold">
    <td>Total de la commande</td>
    <td><?php $total_commande = $total_articles+$frais_livraison; echo $total_commande ?> €</td>
</tr>
</table>
<button class="btn btn-secondary btn-block mt-3"><a style="text-decoration : none;
color:aliceblue;" href="index.php">Encore une envie de nostalgie ?</a></button>
<button class="btn btn-secondary btn-block mt-3" id="nextToStep2">Valider mon panier</button>
</div>
</div>
<!-- Etape Livraison -->
<div id="step2-content" style="display:none;">
    <h4 class="text-uppercase text-pink mb-4">Adresse de livraison</h4>
    <div class="mb-3">
        <label for="nom" class="form-label">Nom</label>
        <input type="text" class="form-control" id="nom" placeholder="Entrez votre nom">
    </div>
    <div class="mb-3">
        <label for="adresse" class="form-label">Adresse</label>
        <input type="text" class="form-control" id="adresse" placeholder="Entrez votre adresse">
    </div>
    <button class="btn btn-secondary btn-block mt-3" id="prevToStep1">Précédent</button>
    <button class="btn btn-secondary btn-block mt-3" id="nextToStep3">Continuer au paiement</button>
</div>

<!-- Etape Paiement -->
<div id="step3-content" style="display:none;">
    <h4 class="text-uppercase text-pink mb-4">Détails de paiement</h4>
    <div class="mb-3">
        <label for="carte" class="form-label">Numéro de carte</label>
        <input type="text" class="form-control" id="carte" placeholder="Entrez votre numéro de carte">
    </div>
    <div class="mb-3">
        <label for="expiration" class="form-label">Date d'expiration</label>
        <input type="text" class="form-control" id="expiration" placeholder="MM/AA">
    </div>
    <div class="mb-3">
        <label for="cvv" class="form-label">CVV</label>
        <input type="text" class="form-control" id="cvv" placeholder="CVV">
    </div>
</div>
```



```
</div>
<button class="btn btn-secondary btn-block mt-3" id="prevToStep2">Précédent</button>
<button class="btn btn-secondary btn-block mt-3" id="nextToStep4">Finaliser la commande</button>
</div>

<!-- Etape Finalisation -->
<div id="step4-content" style="display:none;">
  <h4 class="text-uppercase text-pink mb-4">Confirmation de commande</h4>
  <p>Merci pour votre commande !</p>
  <button class="btn btn-secondary btn-block mt-3" id="prevToStep3">Précédent</button>
</div>
</div>
```

```
<script>
$(document).ready(function() {
  let currentIndex = 0; // Index de l'étape actuelle dans la barre de progression
  const steps = $(".progress-step"); // Sélectionne les étapes de progression
  // Fonction pour aller à l'étape suivante
  function goToNextStep(currentStep, nextStep) {
    $(currentStep).hide(); // Masquer l'étape actuelle
    $(nextStep).show(); // Afficher l'étape suivante
    // Activer l'étape suivante dans la barre de progression
    $(steps.eq(currentIndex)).removeClass("active");
    currentIndex++;
    $(steps.eq(currentIndex)).addClass("active");
  }
  // Fonction pour revenir à l'étape précédente
  function goToPreviousStep(currentStep, previousStep) {
    $(currentStep).hide(); // Masquer l'étape actuelle
    $(previousStep).show(); // Afficher l'étape précédente

    // Activer l'étape précédente dans la barre de progression
    $(steps.eq(currentIndex)).removeClass("active");
    currentIndex--;
    $(steps.eq(currentIndex)).addClass("active");
  }

  // Navigation entre les étapes
  $("#nextToStep2").click(function() {
    goToNextStep("#step1-content", "#step2-content");
  });

  $("#nextToStep3").click(function() {
    goToNextStep("#step2-content", "#step3-content");
  });
  $("#nextToStep4").click(function() {
```



```
    goToNextStep("#step3-content", "#step4-content");
  });
  // Navigation en arrière
  $("#prevToStep1").click(function() {
    goToPreviousStep("#step2-content", "#step1-content");
  });
  $("#prevToStep2").click(function() {
    goToPreviousStep("#step3-content", "#step2-content");
  });

  $("#prevToStep3").click(function() {
    goToPreviousStep("#step4-content", "#step3-content");
  });
});
function checkNegativeOnBlur(input) {
  if (input.value < 1) {
    input.value = 1;
  }
}
function sendInputValue(input) {
  const value = parseInt(input.value, 10);
  if (isNaN(value) || value < 1) {
    input.value = 1;
    return;
  }
  fetch('shoppingCart.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/x-www-form-urlencoded',
    },
    body: `quantity=${encodeURIComponent(value)}`
  })
  .then(response => response.json())
  .then(data => {
    document.getElementById('prix-total').textContent = `${data.total} €`;
    document.getElementById('current-quantity').textContent = data.quantity;
  })
  .catch(error => console.error('Erreur:', error));
}
</script>
```

Figure 3 Code : Panier