

2023-2024

DÉVELOPPEMENT WEB

ING 1 GM Groupe 3



TECH

Iyad BEN MOSBAH

Salim OUESLATI

Charles LIU

Kathirvele RANGANADANE

Roshanth RUBAN

INTRODUCTION

Ce rapport présente le développement de "La Gourmandise", un site web de vente de chocolat en ligne créé dans le cadre du projet en développement web. Une sélection variée de chocolats est disponible sur le site, offrant aux utilisateurs une expérience d'achat à la fois conviviale, intuitive et stimulante.

Le site se distingue par sa structure claire mais néanmoins élaborée, facilitant de manière efficace la navigation des utilisateurs. Le site propose trois catégories principales bien organisées : chocolat noir, chocolat au lait et chocolat blanc, chacune comprenant cinq produits uniques.

Les utilisateurs peuvent ainsi découvrir une variété de produits, allant du chocolat noir intense aux saveurs subtiles et fruitées, en passant par le chocolat au lait crémeux et le chocolat blanc onctueux. Chaque catégorie comprend cinq produits uniques, offrant aux clients une diversité de choix pour répondre à leurs préférences gustatives.

The word "chocolat" is written in a large, elegant, dark brown cursive script. The letters are fluid and connected, with a long, sweeping tail on the final 'e' that extends to the right. The overall style is sophisticated and artistic, typical of branding for a gourmet chocolate brand.

Partie 1 : HTML/CSS

Voici la présentation de notre page d'accueil.

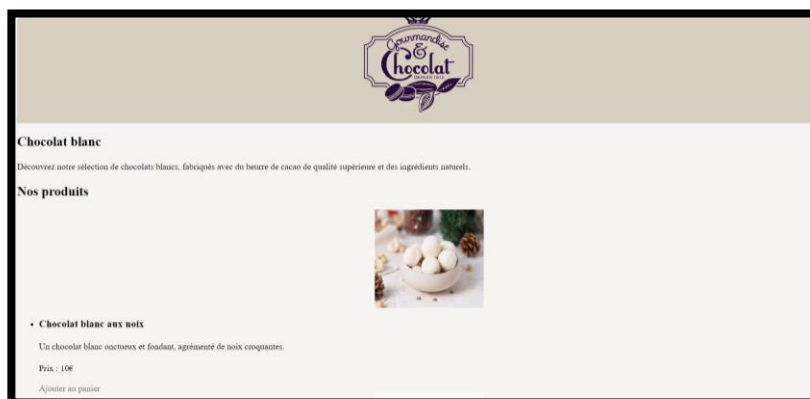


I- Structure HTML (page d'accueil)

La page d'accueil du site utilise une structure HTML bien organisée conforme aux normes du web moderne. Elle est divisée en sections distinctes pour le contenu principal, l'en-tête, le pied de page et la navigation. Le titre "Bienvenue chez La Gourmandise" attire l'attention, avec des descriptions des produits et des catégories de chocolats présentées sous forme de liens. Un paragraphe affiche l'année de création de la chocolaterie (2024), le copyright et un lien vers la page contact.

II- CSS (Style et design)

Le code CSS stylise les éléments HTML, garantissant une présentation professionnelle et accessible. Une palette de couleurs beige et marron est appliquée à l'arrière-plan, aux titres, aux liens et aux autres éléments. Le contenu est centré horizontalement pour une présentation équilibrée sur tous les appareils.



Partie 2 : JavaScript

1-Pour chaque produit

Nous avons ajouté une section quantité et stock sous chaque produit pour guider l'utilisateur. Un bouton "Ajouter au panier" est disponible avec un icône panier. La fonction du bouton (+) permet d'ajouter des chocolats blancs aux noix en incrémentant la quantité, avec des conditions fixant la quantité minimale à 0 et maximale à 20. De plus, l'image peut être agrandie en cliquant dessus.



Le formulaire de contact utilise JavaScript pour vérifier toutes les données saisies. Deux fonctions ont été créées : la première teste le format requis pour chaque champ du formulaire, tandis que la seconde vérifie si tous les champs ont été correctement remplis. Si c'est le cas, les données sont acceptées, sinon une erreur de saisie est affichée en rouge dans le champ spécifique.

Formulaire de contact

N'hésitez pas à nous contacter si vous avez des questions, des suggestions ou si vous souhaitez passer une commande.

Date de contact :

Nom :

Prénom :

Email :

Genre : Veuillez remplir ce champ.

Date de naissance :

Fonction :

Sujet :

Message :

Partie 3 : PHP

1-Factoriser le code html

Nous avons la répétition de plusieurs parties de code html telles que l'en-tête, la fin et la navigation du site. Pour résoudre ce problème, nous avons créé trois fichiers PHP distincts : header.php, footer.php et navigation.php. Ci-dessous se trouve le contenu du fichier navigation.php :

```
factorisation > navigation.php
<header>
  <h1>Chocolaterie La Gourmandise</h1>
  
  <nav>
    <ul>
      <li><a href="../../ACCUEIL.html">Accueil</a></li>
      <li><a href="Blanc.html">Chocolat blanc</a></li>
      <li><a href="Lait.html">Chocolat au lait</a></li>
      <li><a href="noir.html">Chocolat noir</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
  <div class="header-icônes"> <!-- Ajout de l'icone panier -->
    <a href="../../html/panier.html" class="panier-icone">
      
      <span class="panier-texte">Panier</span>
    </a>
  </div>
</header>
```

Nous avons la page factorisationblanc.html dans lequel nous avons mis le code surligné en bleu afin d'ajouter le header.php :

```
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Chocolat blanc - Chocolaterie La Gourmandise</title>
7   <link rel="stylesheet" href="../../css/styleblanc.css">
8   <script src="../../js/script.js" defer></script> <!-- Ajout du script JavaScript -->
9 </head>
10
11 <body>
12   <?php include '../../php/factorisation/navigation.php'; ?> <!-- Inclure le fichier de navigation -->
13
14   <main>
15     <section>
16       <h2>Chocolat blanc</h2>
17       <p>Découvrez notre sélection de chocolats blancs, fabriqués avec du beurre de cacao de qualité supérieure et des ingrédients naturels.</p>
18     </section>
19     <section>
20       <h2>Nos produits</h2>
21       <button id="toggleStock">Stock</button> <!-- Bouton pour afficher/cacher la colonne de la quantité en stock -->
22       <ul id="productlist">
```

On a fait le même procédé pour les autres page html, ce qui nous a permis de réduire les lignes de codes répétitives.

2-Création de session sur chaque page :

On crée un fichier php **varsession.inc.php** contenant les variables de session dont le tableau de catégories avec les produits et les informations.

Voici le code contenu dans le fichier :

```
1  <?php
2  session_start();
3
4  // Tableau des catégories avec les produits et leurs informations
5  $categories = array(
6      'blanc' => array(
7          array(
8              'nom' => 'Chocolat blanc aux noix',
9              'prix' => '10€',
10             'stock' => 20,
11             'image' => 'chocolat_blanc_noix.jpg',
12             'description' => 'Un chocolat blanc onctueux et fondant, agrémenté de noix croquantes.'
13         ),
14         array(
15             'nom' => 'Chocolat blanc aux amandes',
16             'prix' => '12€',
17             'stock' => 20,
18             'image' => 'chocolat_blanc_amande.jpg',
19             'description' => 'Un chocolat blanc délicat et parfumé, avec des amandes entières.'
20         ),
21         array(
22             'nom' => 'Chocolat blanc nature',
23             'prix' => '8€',
24             'stock' => 20,
25             'image' => 'chocolat_blanc_nature.jpg',
26             'description' => 'Un chocolat blanc simple et savoureux, à la saveur pure et authentique.'
27         ),
28         array(
29             'nom' => 'Chocolat blanc à la noisette',
30             'prix' => '9€',
31             'stock' => 20,
32             'image' => 'chocolat_blanc_noisette.jpg',
33             'description' => 'Un chocolat blanc onctueux et gourmand, parsemé d\'éclats de noisettes.'
34         ),
35         array(
36             'nom' => 'Chocolat blanc praliné',
37             'prix' => '11€',
38             'stock' => 20,
39             'image' => 'chocolat_blanc_praline.jpg',
40             'description' => 'Un chocolat blanc irrésistible, avec une ganache praliné onctueuse et fondante.'
41         ),
42         // Ajoutez d'autres produits de la catégorie ici
43     ),
44     // Ajoutez les autres catégories ici
45 );
46 $_SESSION['categories'] = $categories;
47 ?>
```

Pour la création de session dans chaque page on utilise la fonctionnalité **sessionstart()**.

```
<?php
session_start();
```

\$categorie est la variable correspondant au tableau des catégories définie par différents **array** pour chaque catégorie de chocolat avec leurs informations qui leur correspond respectivement.

3- Transfert des informations catégories et produits au format JSON :

```
$_SESSION['categories'] = $categories;

// Convertir en JSON
$jsonData = json_encode($categories);

// Écrire dans un fichier
file_put_contents('categories.json', $jsonData);
```

- Stocker les catégories dans une session :

\$_SESSION['categories'] = \$categories stocke le tableau **\$categories** dans la variable de session **\$_SESSION['categories']**. Cela permet de rendre les catégories disponibles dans d'autres parties du site Web qui ont accès à la session.

- Convertir les informations des catégories en format JSON on utilise la fonction **json_encode()**.

```
// Convertir en JSON
$jsonData = json_encode($categories);
```

3- Ecrire les catégories au formats JSON :

Ceci crée une sauvegarde des catégories au format JSON, qui peut être lue ultérieurement ou utilisée à d'autres fins.

```
// Écrire dans un fichier
file_put_contents('categories.json', $jsonData);
```

Pour le chargement des informations au démarrage du site on a écrit ce code php dans un fichier nommé `demarrage_catégories.php` :

```
<?php
// Charger les données depuis le fichier JSON
$jsonData = file_get_contents('categories.json');

// Décoder les données JSON en tableau associatif
$categories = json_decode($jsonData, true);

// Vérifier si le décodage a réussi
if ($categories === null) {
    die('Erreur de chargement des données.');
}
?>
```

La ligne **\$jsonData = file_get_contents('categories.json')** lit le contenu du fichier **categories.json** et l'enregistre dans la variable **\$jsonData**.

```
// Charger les données depuis le fichier JSON
$jsonData = file_get_contents('categories.json');
```

\$categories = json_decode(\$jsonData, true) convertit la chaîne JSON contenue dans **\$jsonData** en un tableau PHP associatif. Un tableau associatif est un type de tableau en PHP où chaque élément est accessible par une clé (généralement une chaîne de caractères) au lieu d'un index numérique.

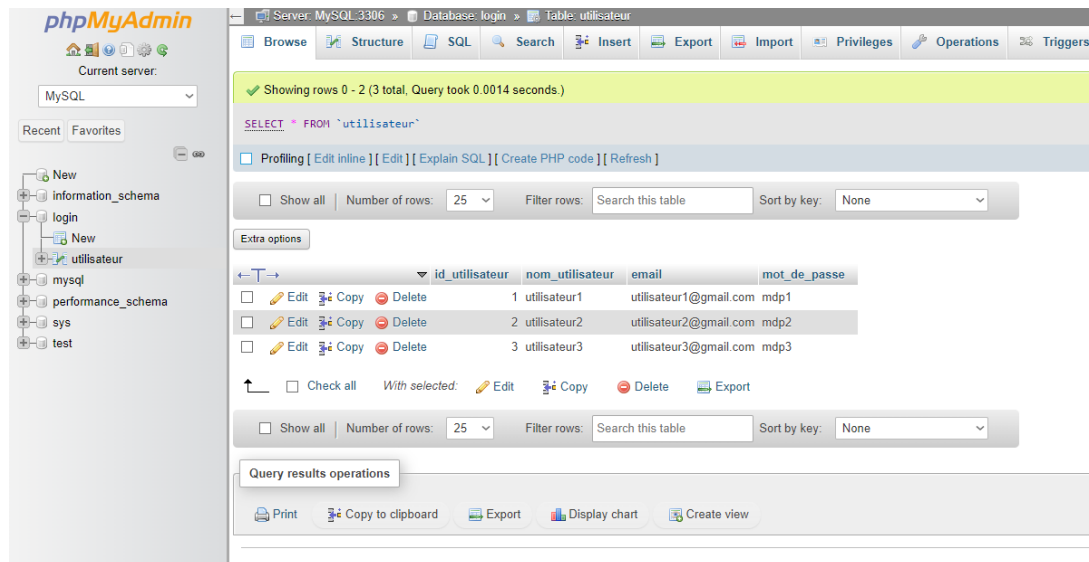
```
// Décoder les données JSON en tableau associatif
$categories = json_decode($jsonData, true);
```

La condition **if** vérifie si le décodage JSON a réussi. La fonction **json_decode()** renvoie **null** en cas d'erreur.

```
// Vérifier si le décodage a réussi
if ($categories === null) {
    die('Erreur de chargement des données.');
}
?>
```

4- Liste des utilisateurs du site :

Dans phpMyAdmin, on a créé une table des utilisateurs du site contenant les informations sur l'email et le mot de passe.



Dans le fichier connexion.php, on peut accéder à la table utilisateur et vérifier si les coordonnées rentrées appartiennent à la base de données :

```

30 <?php
31
32 $hote = "localhost";
33 $nom_base = "login";
34 $utilisateur = "root";
35 $mot_de_passe = "cytech0001";
36
37 try {
38     $conn = new PDO("mysql:host=$hote;dbname=$nom_base", $utilisateur, $mot_de_passe);
39     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
40 } catch(PDOException $e) {
41     echo "Échec de la connexion : " . $e->getMessage();
42     die(); // Arrêter l'exécution en cas d'erreur de connexion
43 }
44
45 // Démarrage de la session pour stocker les informations utilisateur
46 session_start();
47
48 // Traitement du formulaire de connexion
49 if ($_SERVER["REQUEST_METHOD"] == "POST") {
50     $nom_utilisateur = trim($_POST["username"]);
51     $mot_de_passe = trim($_POST["password"]);
52
53     // Préparer la requête SQL pour récupérer l'utilisateur par nom d'utilisateur
54     $sql = "SELECT * FROM utilisateur WHERE nom_utilisateur = :nom_utilisateur";
55     $stmt = $conn->prepare($sql);
56     $stmt->bindParam(":nom_utilisateur", $nom_utilisateur);
57     $stmt->execute();
58
59     $utilisateur = $stmt->fetch(PDO::FETCH_ASSOC);
60
61     if ($utilisateur && ($mot_de_passe==$utilisateur["mot_de_passe"])) {
62         // Utilisateur authentifié, stocker les informations dans la session
63         $_SESSION["id_utilisateur"] = $utilisateur["id"];
64         $_SESSION["nom_utilisateur"] = $nom_utilisateur;
65         header("Location: ../Accueil.html"); // Redirection vers la page d'accueil
66         die(); // Arrêter l'exécution après la redirection
67     } else {
68         echo "<p class='erreur'>Erreur de connexion, nom d'utilisateur ou mot de passe incorrect.</p>";
69     }
70 }
71
72 >?

```

L'hôte est ici localhost, car on a créé la table utilisateur dans phpMyAdmin sur Wamp en localhost. Il faudrait sûrement le partager.

La variable **utilisateur** la valeur false si l'utilisateur rentré n'est pas dans la base de données, sinon elle est true est prend sa valeur.

5- Formulaire de contact

Nous avons fais des fonctions afin de controler les éléments de la page contact.

1. **addToCart(\$productName, \$quantity, \$price)** : Ajoute un produit au panier. Il récupère les données du produit, vérifie s'il est déjà dans le panier, met à jour la quantité ou l'ajoute au panier, puis met à jour le panier dans le stockage de session.
2. **validateForm()** : Valide les données d'un formulaire de contact. Il vérifie si les champs nom, email et message sont renseignés correctement. Il retourne true si toutes les validations sont réussies, sinon false.

```
<?php
// Fonction pour ajouter le produit au panier
function addToCart($productName, $quantity, $price) {
    // Créer un objet représentant le produit à ajouter au panier
    $quantity = intval($_POST['quantity']);

    $product = array(
        'name' => $productName,
        'quantity' => $quantity,
        'price' => $price
    );

    // Récupérer le panier depuis le stockage local s'il existe, sinon initialiser un panier vide
    $cart = json_decode($_SESSION['cart'], true) ?: array();

    // Vérifier si le produit est déjà dans le panier
    $existingProductIndex = array_search($product['name'], array_column($cart, 'name'));

    if ($existingProductIndex !== false) {
        // Si le produit existe déjà dans le panier, mettre à jour la quantité
        $cart[$existingProductIndex]['quantity'] += $product['quantity'];
    } else {
        // Sinon, ajouter le produit au panier
        $cart[] = $product;
    }

    // Mettre à jour le panier dans le stockage local
    $_SESSION['cart'] = json_encode($cart);

    // Afficher un message de confirmation ou mettre à jour l'interface utilisateur
    echo 'Le produit a été ajouté au panier !';
}
```

Partie 4 : PHD-BDD-AJAX

1- Parcours des fichiers HTML

Le code utilise une boucle foreach pour parcourir chaque fichier HTML (blanc.html, noir.html, lait.html). Cela permet de traiter chaque fichier indépendamment et d'extraire les données des produits correspondants.

```
<?php

// Process each HTML file (blanc.html, noir.html, lait.html)
foreach (['blanc.html', 'noir.html', 'lait.html'] as $fileName) {
    $htmlContent = file_get_contents($fileName);
    $document = new DOMDocument();
    $document->loadHTML($htmlContent);

    // Extract product information from the DOM object
    $listeProduits = $document->getElementsByTagName('li');
    foreach ($listeProduits as $produitElement) {
        $nomProduit = $produitElement->getElementsByTagName('h3')[0]->textContent; // Assuming product name is in the first 'h3' tag
        $prixProduit = $produitElement->getElementsByTagName('p')[1]->textContent; // Assuming product price is in the second 'p' tag

        // Extract product category from the file name
        $categorieProduit = ucfirst(str_replace('html', '', $fileName)); // Capitalize and remove 'html' from filename

        // Store product data in the array
        $productData[] = [
            'nom' => $nomProduit,
            'prix' => $prixProduit,
            'categorie' => $categorieProduit
        ];

        // Add category to the categories array if not already present (optional, depends on your logic)
        if (!in_array($categorieProduit, $categories)) {
            $categories[] = $categorieProduit;
        }
    }
}

// Include the file containing database functions (assuming it's named bdd.php)
require_once('bdd.php');

// Call the function from bdd.php to insert the extracted product data
insertProducts($productData);

echo "Données insérées dans la base de données avec succès!";
```

2. Extraction des informations produit

A l'intérieur de la boucle foreach, le code utilise des techniques de manipulation de DOM (Document Object Model) pour analyser le contenu HTML de chaque fichier. Il cible spécifiquement les éléments HTML pertinents pour extraire les informations produit.

- **Nom du produit:** Il est généralement extrait d'un titre (<h1>, <h2>, etc.) ou d'un élément dédié au nom du produit.
- **Prix du produit:** Il est souvent trouvé dans un paragraphe ou un élément dédié au prix.
- **Catégorie du produit:** Elle peut être déduite du nom du fichier lui-même (ex: "blanc.html" -> catégorie "blanc").

3. Stockage des données dans un tableau

Les informations extraites pour chaque produit sont stockées dans un tableau associatif nommé \$productData. Ce tableau permet de regrouper les données de manière structurée et d'accéder facilement aux informations individuelles des produits.

4. Gestion des catégories

- **Extraction des catégories:** Le code peut utiliser une fonction getAllCategories() (supposée existante) pour récupérer les catégories déjà présentes dans la base de données.
- **Ajout de nouvelles catégories:** Si une catégorie déduite d'un nom de fichier n'est pas encore présente dans la base de données, elle est ajoutée à la liste des catégories (\$categories).
- **Association produit-catégorie:** L'ID de la catégorie correspondante est associé à chaque produit dans le tableau \$productData

AJAX

```
<?php

// Include database connection variables
require_once('bddData.php');

// Function to update product stock
function updateProductStock($productID, $quantityChange) {
    if (!empty($productID) && is_numeric($quantityChange)) {
        $db = connectDB();

        // Prepare and execute the update query (replace with your actual query)
        $updateQuery = $db->prepare("UPDATE produits SET stock = stock + ? WHERE id = ?");
        $updateQuery->bind_param('ii', $quantityChange, $productID);
        $updateQuery->execute() or die('Error updating stock: ' . $db->error());

        disconnectDB($db);

        return true;
    } else {
        return false;
    }
}

// Check if AJAX request is valid
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $productID = intval($_POST['productID']);
    $quantityChange = intval($_POST['quantity']);

    $success = updateProductStock($productID, $quantityChange);

    $response = [
        'success' => $success,
        'error' => $success ? null : 'Error updating stock'
    ];

    echo json_encode($response);
} else {
    // Handle invalid request
    echo json_encode(['success' => false, 'error' => 'Invalid request']);
}
```

Mise à jour du stock des produits (fichier séparé)

- Ce fragment de code définit une fonction updateProductStock qui met à jour le stock d'un produit en fonction de son ID et d'une variation de quantité.
- Il utilise des requêtes préparées pour une interaction sécurisée avec la base de données.
- Il vérifie si la requête est valide (POST) et retourne une réponse JSON indiquant le succès ou l'erreur.

Conclusion

En conclusion, ce projet de développement web a été une expérience enrichissante pour notre groupe, nous permettant d'approfondir nos connaissances dans plusieurs domaines clés. Tout au long du processus, nous avons eu l'opportunité de mettre en pratique nos compétences en HTML, CSS, JavaScript et PHP pour concevoir et implémenter un site web fonctionnel et attrayant.

L'utilisation de différentes technologies nous a permis de créer une expérience utilisateur conviviale et intuitive sur "La Gourmandise", notre site de vente de chocolat en ligne. Nous avons développé des fonctionnalités telles que la gestion de panier avec JavaScript, la validation de formulaire avec PHP, la factorisation du code avec l'utilisation de fichiers PHP séparés pour l'en-tête, le pied de page et la navigation, et l'intégration de données dynamiques à partir d'une base de données MySQL.

En outre, ce projet nous a également confrontés à des défis intéressants, notamment la manipulation de DOM pour extraire des informations à partir de fichiers HTML, la gestion des sessions et des cookies pour le suivi des utilisateurs et la sécurisation des interactions avec la base de données.

Dans l'ensemble, cette expérience nous a permis de consolider nos compétences techniques et de développer notre compréhension des meilleures pratiques en développement web. Nous sommes fiers du résultat obtenu et confiants dans notre capacité à appliquer ces connaissances dans des projets futurs.

Merci pour la lecture de ce rapport

Lien de la vidéo : <https://watch.wave.video/vvB25uNqp2x6RFeP>