



Fiche de production

Situation E4

Mise en place d'un système de panier électronique

Fiche E4

Session 2019-2020

Dans le cadre de la réalisation d'un site de vente en ligne, le projet Dendō jitensha, il nous faut mettre en place d'un système de panier. Pour permettre à l'utilisateur de pouvoir acheter ses articles en une fois et aussi de lui permettre de voir le prix total de sa commande.

Cette fiche de production a pour objectif de présenter le procédé permettant la mise en place de cette fonctionnalité au sein de l'application Web développée.

Pour permettre une présentation complète de la situation, plusieurs points seront abordés :

- Présentation de manière globale de la situation de projet ;
- Présentation fonctionnelle de la situation ;
- Mise en place de tests permettant de démontrer le bon fonctionnement des modules.

Le dossier ci-présent est étroitement lié à la situation E4 nommée « Mise en place d'un système d'inscription et de connexion d'un utilisateur ». Aussi, certaines parties du dossier, si elles ne sont pas claires, feront références au dossier de cette autre situation.

Il est aussi important de noter que le projet est hébergé sur le compte GitHub de Dam's-debug

Les différents sujets abordés seront revus de manière plus approfondie dans la table des matières présente sur la page suivante.

Table des matières

I. CONTEXTE	P.1-3
1a. Description de la situation	p.1
1b. Attentes du projet	p.1-2
1c. Définitions des besoins.....	p.2-3
II. PRESENTATION DE LA PARTIE FRONT END	P.3-5
2a. Définition des éléments de la partie visuel du projet	p.3
2b. Page de visualisation des produits	p.4-5
2c. Panier électronique	p.5
III. PRESENTATION DE LA PARTIE BACK END	P.6-15
3a. Présentation de l'architecture de la base de données	p.6-8
3b. Présentation du script de création de la base de données	p.9-10
3c. Description du système d'ajout au panier	p.10-12
3d. Affichage des informations des articles ajoutés	p.13-14
3e. Définition du prix total des articles	p.14
3f. Suppression d'un article	p.15
IV. TEST DU SYSTEME MIS EN PLACE	P.16-17
4a. Test du système d'ajout au panier (utilisateur connecté)	p.15
4b. Test du système d'ajout au panier (utilisateur non-connecté)	P.16
CONCLUSION.....	P.16

I- Contexte

Cette première partie a pour but de contextualiser la situation rencontrée dans le cadre la réalisation de ce projet E4.

1a. Description de la situation

La société Dendō jitenasha, acteur majeur sur le marché asiatique de la vente de vélo électrique souhaite étendre son activité en intégrant le marché français. La solution est de mettre en place des solutions numériques qui permettront d'accroître la visibilité de la société et d'attirer une nouvelle clientèle, en partie grâce au développement d'une application Web. Il s'agira plus précisément d'un site de vente en ligne (e-commerce), qui permettra la présentation des produits vendus par Dendō jitenasha et bien sûr la possibilité d'achat par les utilisateurs.

1b. Attentes du projet

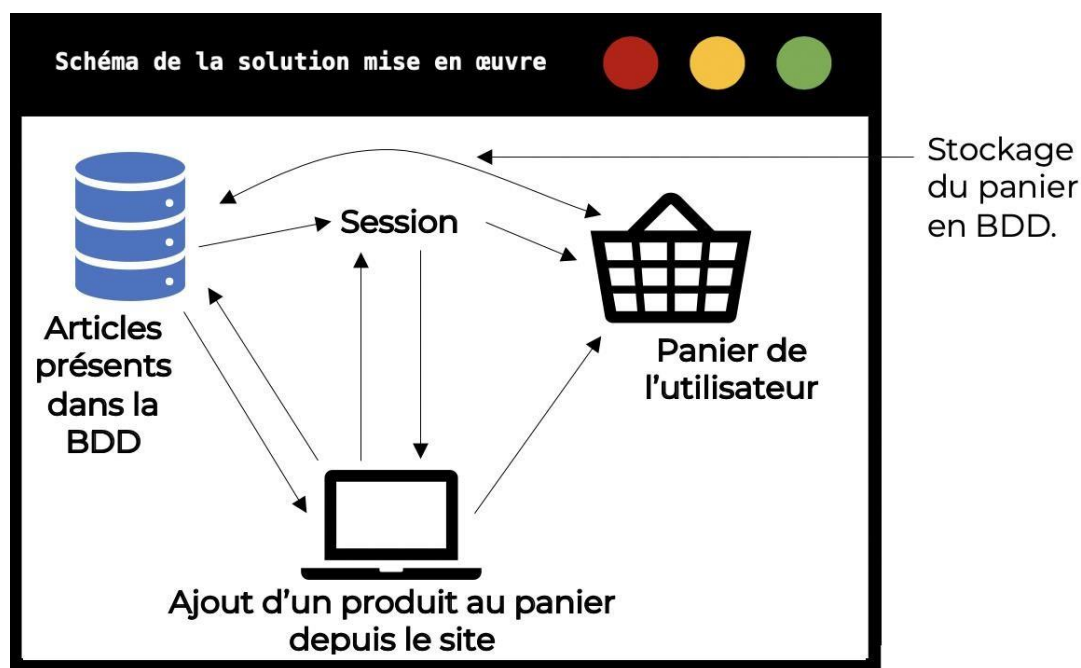
Dans un premier, il est nécessaire de préciser que le fonctionnement du système de panier diffèrera légèrement en fonction du statut de l'utilisateur. En effet, on distinguera 2 types d'utilisateurs :

- les utilisateurs connectés ;
- les utilisateurs non-connectés.

Le système mise en place devra permettre à un utilisateur, en fonction de son statut, de réaliser les actions suivantes :

Système de panier électronique	
Statut de l'utilisateur	
Utilisateur connecté	Utilisateur non-connecté
Ajout d'un ou plusieurs article(s) au panier.	Ajout d'un ou plusieurs article(s) au panier.
Visualisation du ou des article(s) ajouté(s) dans le panier.	Visualisation du ou des article(s) ajouté(s) dans le panier.
Visualiser les prix individuels des articles en fonctions du nombre de pièce(s).	Visualiser les prix individuels des articles en fonctions du nombre de pièce(s).
Visualisation des prix totaux HT et TTC d'un panier.	Visualisation des prix totaux HT et TTC d'un panier.
Supprimer un article de son panier.	Supprimer un article de son panier.
Le panier est stocké en base de données, donc le panier ne sera pas supprimé lorsque l'utilisateur se déconnectera ou fermera son navigateur Web.	Le panier n'est stocké qu'en session, donc le panier sera supprimé lorsque l'utilisateur fermera son navigateur Web.

Il est possible de visualiser ces différents points à l'aide de la figure ci-dessous :



Comme indiqué plutôt dans le dossier, le panier constituera un espace de stockage « tampon » avant paiement des articles. Cette fonctionnalité devra donc être implémentée. La mise en place du système de paiement ne sera pas présentée dans ce dossier.

1c. Définitions des besoins

De manière à permettre la mise en place d'un tel système, il sera nécessaire de mettre en place un environnement de développement sur lequel sera installés un langage serveur permettant la gestion de la partie back end (dans notre cas, le langage choisi est le langage PHP + utilisation du gestionnaire de librairie nommé Composer de manière à mettre en place un système d'autoloading performant).

Un Système de Gestion de Base de Données, autrement appelé SGBD, de manière à mettre en place la base de données qui permettra le stockage des paniers des utilisateurs connectés et qui permettra également de manipuler les données stockées par le biais de requêtes SQL (qui devra donc également être présent sur l'environnement).

Enfin, la présence d'un serveur Web (autrement appelé serveur http) sera indispensable dans le cadre de la réalisation des tests de la solution, dans notre cas, il s'agira d'un serveur Apache.

D'autres langages seront également utilisés dans la partie front end cette fois tel que le HTML5, le CSS3 et du JavaScript. Il s'agit ici de langages interprétés par les navigateurs Web (Mozilla Firefox, Google Chrome, Safari, etc...), il faudra donc avoir accès à au moins l'un d'entre eux de manière à pouvoir tester la solution dans son ensemble.

Il sera possible en fonction du système d'exploitation utilisé, d'installer un serveur LAMP, MAMP ou encore WAMP de manière à avoir accès à ces différents langages et outils.

II- Présentation de la partie front end

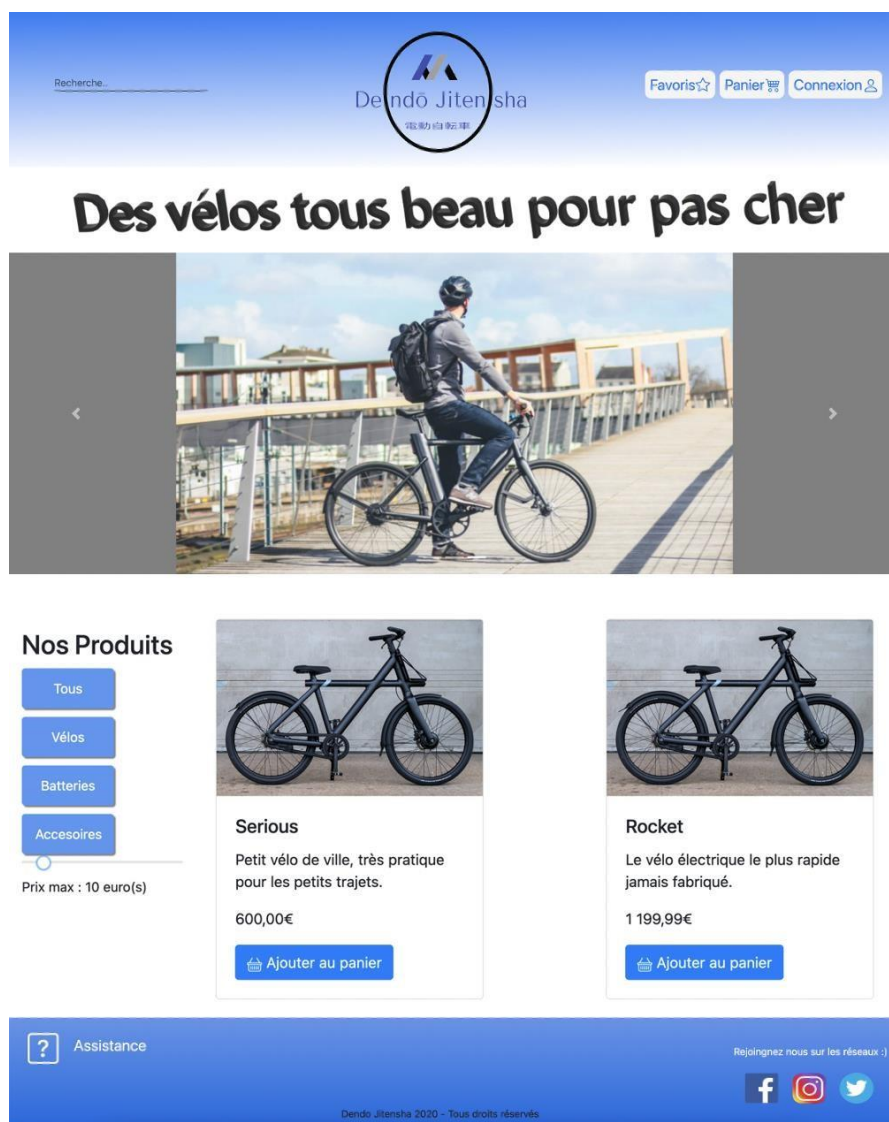
Cette partie permettra de réaliser une présentation de la partie visuelle du projet par le biais de maquettes et ainsi présenter une interface graphique qui se rapprochera du résultat final.

2a. Définition des éléments de la partie visuelle du projet

Pour permettre la mise en place d'un tel projet, il faudra développer les pages suivantes : •
page de présentation des articles + ajout au panier ;
• page panier électronique + suppression d'articles.

2b. Page de visualisation des produits

Dans cette partie vous sera présenté la page de visualisation des articles. L'objectif de cette section est de vous présenter la manière dont les articles seront affichés sur le site via des miniatures.



Les produits seront donc, comme le présente la figure ci-dessus, affichés de manière réduite sur la page de présentation des articles du site.

Un système de carte sera utilisé, et les informations suivantes seront affichés :

- nom de l'article ;
- description de l'article ;
- prix de l'article ;
- bouton d'ajout au panier, qui permettra donc d'ajouter l'article au panier.

Comme vous pouvez le constater, d'autres éléments sont présents sur cette page :

- une barre de navigation ;
- un carrousel ;
- une section de filtre.
- ...

2c. Panier électronique

L'interface graphique du panier électronique sera la suivante :

	Serious Petit vélo de ville, très pratique pour les petits trajets. Quantité : 2	Prix : 600,00€
	Rocket Le vélo électrique le plus rapide jamais fabriqué. Quantité : 1	Prix : 1 199,99€
Total HT: 2 399,99€ Total TTC: 2 879,99€		

Page produits

Comme il est possible de le constater sur la figure ci-dessus, le panier permettra l'affichage des éléments suivants :

- Une ligne par article ;
- Une croix permettant la suppression d'un article ;
- L'image de l'article ;
- Le nom de l'article ;
- La description de l'article ;
- La quantité d'articles identiques sélectionnés ;
- Le prix unitaire des articles ;

Sous cette section de présentation des articles, les éléments suivants sont visibles :

- Le prix total HT ;
- Le prix total TTC.

Il sera également possible de retourner vers la page de présentation des produits, en cliquant sur le bouton « Page produits ».

Lors de l'implémentation du système de paiement, la partie front end du panier sera amené à évoluer avec l'apparition d'un bouton valider mon panier qui mènera vers le système de paiement du site.

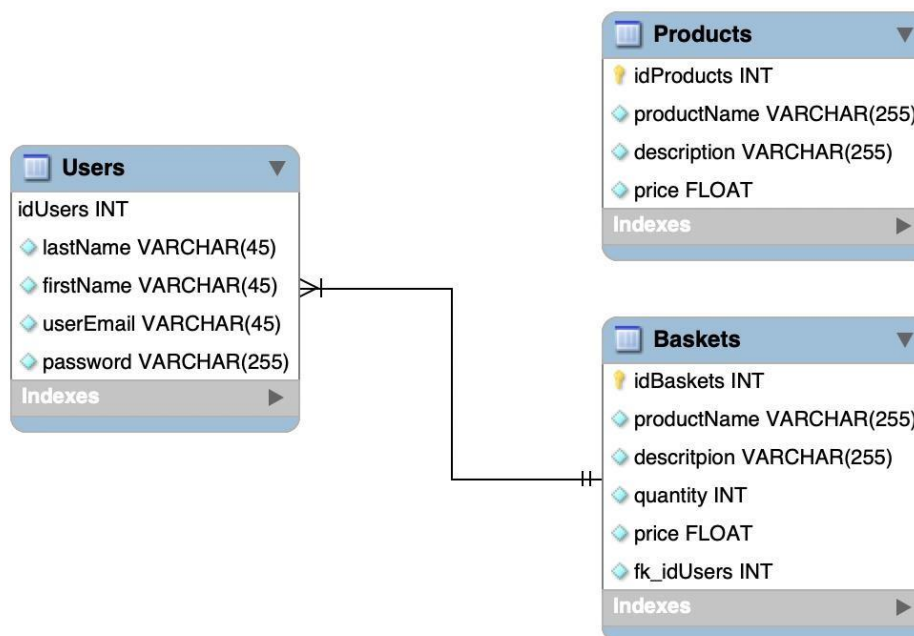
III- Présentation de la partie back end

Dans cette partie du dossier, je vous propose de vous présenter la base de données qui sera mise en place pour permettre le bon fonctionnement du système de panier. Cette section permettra donc de présenter la partie base de données qui comprend donc l'architecture de cette dernière et le script de création de cette même base de données.

Ensuite, il sera important de lier le jeu de données créé avec l'interface utilisateur, par le biais de la mise en place des scripts PHP, qui permettront l'ajout de données pour les sauvegarder de manière persistante. Il sera également possible de manipuler les données stockées et ainsi permettre d'afficher les articles, et les différentes informations liées à ces derniers, dans le panier en fonction de l'utilisateur, s'il est connecté.

3a. Présentation de l'architecture de la base de données

Dans le cadre de cette situation, la mise en place de la base de données nommée « E4_ESPACES_UTILISATEURS » sera nécessaire (cette table correspond à la table créée dans la seconde situation E4 intitulée « Mise en place d'un système d'inscription et de connexion d'un utilisateur »). Elle contiendra les informations suivantes :



Dans cette base de données, la mise en place de 3 tables sera nécessaire :

- Users, qui contiendra les informations des utilisateurs inscrits ;
- Baskets, qui contiendra les informations des articles ajoutés au panier par un utilisateur connecté ;
- Products, qui contiendra les informations des produits du site. *Voir plus de détails dans les différents tableaux suivants.*

Users	
Nom du/des champ(s)	Description
idUsers	Permettra d'identifier un utilisateur en fonction d'un identifiant donné.
lastName et firstName	Permettront de stocker le nom et le prénom des utilisateurs. Ces informations seront utilisées dans de nombreux modules du site tel que l'affichage des informations personnelles d'un utilisateur sur son espace, la mise en place des informations de paiement, de facturation, de livraison, ...

userEmail	Permettra le stockage de l'adresse email d'un utilisateur. Ce champ sera utilisé pour permettre l'identification d'un utilisateur lors de sa connexion. Il s'agira également d'une information qui permettra de savoir si un utilisateur et déjà inscrit sur le site ou non (au moment de son inscription) pour éviter la duplication de comptes.
password	Qui stockera de manière sécurisée (chiffrement) le mot de passe que l'utilisateur aura choisi. Il s'agira de la seconde information (avec l'adresse email) qui sera demandée au moment de la connexion de l'utilisateur de manière à vérifier les informations de ce dernier.

Baskets	
Nom du/des champ(s)	Description
idBaskets	Permettra d'identifier un ou plusieurs élément(s) d'un panier en fonction d'un identifiant donné.
productName	Permettra de stocker le nom du produit ajouté au panier.
description	Permettra de stocker la description d'un produit ajouté au panier.
quantity	Ce champ permettra de stocker la quantité sélectionnée d'un produit donné (si un même article a été choisi plusieurs fois, cette valeur sera incrémentée en fonction du nombre d'article ajouté).
price	Dans ce champ, le prix unitaire des produits ajoutés sera stocké.
fk_idUsers	Ce champ permettra de lier le contenu d'un panier avec un utilisateur inscrit, l'identifiant de l'utilisateur sera stocké ici.

Products	
Nom du/des champ(s)	Description
idProducts	Ce champ permettra d'identifier les articles dans la base de données en fonction d'un numéro identifiant.
productName	Permettra de stocker le nom du produit en base de données.
description	Permettra de stocker la description d'un produit en base de données.
price	Dans ce champ, le prix unitaire des produits ajoutés sera stocké.

Maintenant que la structure de la base de données a été présentée avec précision, la partie suivante fera l'objet de la présentation du script SQL de création de cette même base.

3b. Présentation du script de création de la base de données

Pour permettre la mise en place de la base de données qui sera utilisée par le module consacré à la création d'un système de panier, le script SQL suivant sera utilisé :

```
DROP DATABASE E4_PANIER;
CREATE DATABASE E4_PANIER;
USE E4_PANIER;

CREATE TABLE IF NOT EXISTS Users(  idUsers INT
NOT NULL AUTO_INCREMENT,  lastName
VARCHAR(45) NOT NULL,  firstName VARCHAR(45)
NOT NULL,  userEmail VARCHAR(45) NOT NULL,
password VARCHAR(255) NOT NULL,
PRIMARY KEY (idUsers)
);

CREATE TABLE IF NOT EXISTS Baskets(  idBaskets INT
NOT NULL AUTO_INCREMENT,  productName
VARCHAR(255) NOT NULL,  description VARCHAR(255)
NOT NULL,
quantity INT NOT NULL,  price
FLOAT NOT NULL,  fk_idUsers INT
NOT NULL,  PRIMARY KEY
(idBaskets),
FOREIGN KEY (fk_idUsers) REFERENCES Users(idUsers)
);

CREATE TABLE IF NOT EXISTS Products(  idProducts
INT NOT NULL AUTO_INCREMENT,  productName
VARCHAR(255) NOT NULL,  description
VARCHAR(255) NOT NULL,  price FLOAT NOT NULL
);
```

Il sera possible d'insérer un jeu de données, contenant 2 produits pour la réalisation des tests, en utilisant le script suivant :

```
INSERT INTO `Products` (`productName`, `description`, `price`)
VALUES ('Serious', 'Petit vélo de ville, très pratique pour les petits trajets.', '600'), ('Rocket', 'Le vélo électrique le plus rapide jamais fabriqué.', '1199.99');
```

3c. Description du système d'ajout au panier

Dans un premier il faut savoir les utilisateurs du site pourront avoir des statuts différents. En effet, il y'en aura 2 :

- Connecté, ce statut indique qu'un utilisateur possède un compte sur le site et qu'il est connecté à son espace personnel ;
- Non-connecté, ce statut indique que l'utilisateur n'est pas connecté à un espace utilisateur enregistré en base de données.

Cette information est très importante car elle va permettre de définir 2 systèmes d'ajouts d'un article au panier.

Si un utilisateur possède un espace personnel et qu'il est connecté à ce dernier au moment de l'ajout de l'article au panier, une entrée correspondante au produit ajouté sera créée dans la table « Baskets » et son champ *fk_idUsers* prendra la valeur de l'identifiant de l'utilisateur qui vient d'ajouter l'article. Si un même article est ajouté plusieurs fois au panier, une requête permettra la modification du champ *quantity* de l'entrée du produit dans la table « Baskets » en fonction d'un utilisateur données, ce qui évitera la création d'une nouvelle ligne pour un même produit d'un même utilisateur (dans le cas où le dit produit a déjà été ajouté au moins une fois).

Dans le cas où l'utilisateur ne serait pas connecté (inscrit et pas connecté, ou pas inscrit et donc pas connecté), le panier de l'utilisateur sera uniquement stocké en session. Donc les seules interactions avec la base de données permettront de récupérer les informations des produits qui seront ajoutés au panier stocké dans la session en cours. La fermeture du navigateur de l'utilisateur entraînera la suppression des articles présents dans le panier stocké en session.

Maintenant que ces points ont été éclaircis, voyons en détails le fonctionnement de l'ajout d'un article au panier.

Le lien suivant vous présente la page qui permet de visualiser les produits : https://github.com/Damsdebug/E4_Panier/blob/main/products.php.

Dans ce fichier, les produits sont affichés via une instanciation de la classe PDO qui prend en arguments les informations de connexion à la base de données. Cet objet sera ensuite passé en argument au moment de l'appel de la classe **DbManager**, qui permettra de réaliser un certain nombre de manipulation auprès de la base de données. Cette classe sera utilisée en partie pour récupérer les informations des produits stockés dans la table *Products*, ce qui permettra de réaliser différents traitements. Dans le cadre de l'affichage des produits sur la page de présentation des articles, le résultat d'une requête de sélection de toutes les entrées de la tables *Products*, sera utilisé. Une itération sera réalisée via l'utilisation d'une boucle « foreach » et permettra l'affichage des articles présents en BDD (base de données).

```
<a href="php/addToBasket.php?idProduct=<?= $product['idProducts'] ?>" class="btn btn-  
primary"> Ajouter au panier </a>
```

Le lien ci-dessus sera donc suivi au moment du clique de l'utilisateur connecté sur ce dernier. Il est possible de voir que l'identifiant du produit à ajouter est passé dans l'URL, de manière à transmettre cette information via la méthode « GET » vers le fichier *addToBasket.php*.

Le même code sera exécuté, que l'utilisateur soit connecté ou non. La seule différence entre les deux cas, est qu'un utilisateur connecté verra apparaître le bouton déconnexion, alors qu'un utilisateur non-connecté verra apparaître un bouton de connexion.

Au moment de l'ajout d'un au panier, c'est le fichier *addToBasket.php* (https://github.com/Damsdebug/E4_Panier/blob/main/php/addToBasket.php) qui permettra de réaliser des traitements en fonction du statut de l'utilisateur.

Comme vous pouvez le constater, un bloc conditionnel principal constitue le fichier :

- si une session « \$_SESSION['userEmail'] » existe, alors l'utilisateur est connecté à un espace personnel. Dans cette situation, les éléments ajoutés au panier seront sauvegardés de manière persistante dans la base de données ;
- si la session « \$_SESSION['userEmail'] » n'existe pas, alors les informations des produits ajoutés au panier seront stockées en session et non en dans la base de données.

Dans les deux situations citées ci-dessus, des tests sont réalisés de manière à savoir un produit à bien été sélectionné et savoir si le produit que l'utilisateur souhaite ajouter à son panier est bien présent de BDD.

Si un utilisateur connecté ajoute un article à son panier électronique, c'est la classe **DbManager** (https://github.com/Dams-debug/E4_Panier/blob/main/php/class/Database/DbManager.php) qui sera utilisée et plus particulièrement sa méthode *add()*. Cette méthode va permettre la réalisation de plusieurs actions :

- vérifier si un article similaire est déjà présent dans le panier de l'utilisateur ;
 - si l'article n'est pas encore présent dans le panier de l'utilisateur, le produit est simplement ajouté au panier de l'utilisateur (ajout d'une entrée à la table *Baskets* ;
 - si l'article est déjà présent dans le panier, seule la quantité du produit sera modifiée dans le panier (si l'utilisateur avait déjà un exemplaire de l'article qu'il souhaite ajouter dans son panier, la valeur quantité passera de 1 à 2.

Si un utilisateur non-connecté souhaite ajouter un article à son panier, c'est la classe **BasketManager** (https://github.com/Damsdebug/E4_Panier/blob/main/php/class/BasketHandling/BasketManager.php).

Au moment de l'instanciation de cette classe, une variable « \$_SESSION['basket'] » sera créée. Elle constituera le panier stocké en session sous forme d'un tableau associatif et chaque index correspondra à un article.

Dans le cadre de l'ajout d'un élément au panier, c'est la méthode *add()* de cette même classe qui permettra l'ajout de l'article. Elle effectue les actions suivantes :

- si un indice ayant pour valeur l'identifiant du produit à ajouter est déjà présent, cela signifie qu'un article similaire a déjà été ajouté. À ce moment-là, la valeur de cet index est incrémentée de 1 (cela correspond à la quantité du produit) ;

- si au contraire, l'indice du produit à ajouter n'existe pas encore dans le tableau, il faut l'ajouter en créant l'index et en lui donnant pour valeur 1.

3d. Affichage des informations des articles ajoutés

C'est le fichier basket.php (https://github.com/Dams-debug/E4_Panier/blob/main/basket.php) qui permettra la visualisation des produits présents dans le panier.

Le statut de l'utilisateur définira la manière dont les articles seront affichés. En effet, si un utilisateur n'est pas connecté, c'est par le biais de la session encours que les articles et leurs informations seront affichés.

```
// On récupère les indices présent dans la session qui a le nom 'basket'
$tmpIds = array_keys($_SESSION['basket']);
$ids = implode(",",$tmpIds);
if (empty($ids)) {
    $products = array();
} else {
    $products = $dbConnection->query("SELECT * FROM Products WHERE idProducts
IN (". $ids. ")");
}
```

Le bloc de code ci-dessus, permet de récupérer les informations des produits stockés dans le panier en session en fonction des index présent dans le tableau (*\$dbConnection* est une instance de la classe **DbManager**).

Comme l'indique la figure suivante, c'est une boucle « foreach » qui permettra l'affichage des produits :

```

foreach ($products as $product) {
    ?>
    <div class="basket-product">
        <div class="product-delete">
            <a href="basket.php?del2=<?=$product['idProducts'];?>"> <h3>X</h3> </a>
        </div>
        <div class="product-img">
            
        </div>
        <div class="product-description">
            <h2> <?=$product['productName']; ?> </h2>
            <p> <?=$product['description']; ?> </p>
            <p> Quantité : <?=$SESSION['basket'][$product['idProducts']]; ?></p>
        </div>
        <div class="product-price">
            <p>Prix : <?= number_format($product['price'],2,',',' ') .'€'; ?></p>
        </div>
    </div>
    <?php
}

```

Si l'utilisateur est connecté, alors les produits et leurs informations seront directement récupérées dans la base de données (voir figure suivante).

```

$products = $dbConnection->query("SELECT * FROM Baskets WHERE fk_idUsers =". $SESSION['idUser'] ."" );
$price = 0;
foreach ($products as $product) {
    ?>
    <div class="basket-product">
        <div class="product-delete">
            <a href="basket.php?del=<?=$product['productName'];?>&del1=<?=$product['fk_idUsers'];?>"> <h3>X</h3> </a>
        </div>
        <div class="product-img">
            
        </div>
        <div class="product-description">
            <h2> <?=$product['productName']; ?> </h2>
            <p> <?=$product['description']; ?> </p>
            <p> Quantité : <?=$product['quantity']; ?></p>
        </div>
        <div class="product-price">
            <p>Prix : <?= number_format($product['price'] * $product['quantity'],2,',',' ') .'€'; ?></p>
            <?php $price += $product['price'] * $product['quantity']; ?>
        </div>
    </div>
    <?php
}

```

3e. Définition du prix total des articles

Le prix des produits affichés dans le panier sont les prix unitaire des produits multipliés par la quantité de produits. Sous la section qui présente les produits, sont présents 2 prix totaux :

- le prix total HT (Hors Taxes) ;
- le prix TTC (Toutes Taxes Comprises) qui correspond au prix HT auquel s'ajoute une taxe de 20%.

Dans le cas où l'utilisateur n'est pas connecté, le prix total est calculer grâce à la méthode *total()* de la classe **BasketManager**. Via l'utilisation de cette méthode, le prix total sera renvoyé et contiendra le prix d'un article (valeur récupérée en BDD) multiplié par le nombre des différents articles du panier.

Dans le cas où l'utilisateur est connecté à un espace personnel, la variable *\$price*, visible à la seconde ligne de la figure précédente, sera incrémentée par la valeur de la variable *\$product['price']* du produit multiplié par la quantité du produit.

À ce stade le prix HT a été défini, il ne reste plus qu'à multiplier cette valeur par 1,2 de manière à obtenir le montant TTC (dans les deux cas cités juste avant) :

```
<h2>Total <b>HT</b>: <?= number_format($basket->total(),2,',',' ') . '€'; ?></h2>
<h2>Total <b>TTC</b>: <?= number_format($basket->total() * 1.2,2,',',' ') . '€'; ?></h2>
```

Si un utilisateur n'est pas connecté (*\$basket* est une instance de *BasketManager*).

```
<h2>Total <b>HT</b> <?= number_format($price,2,',',' ') . '€'; ?></h2>
<h2>Total <b>TTC</b> <?= number_format($price * 1.2,2,',',' ') . '€'; ?></h2>
```

Si un utilisateur est connecté.

3f. Suppression d'un article

Pour supprimer un article du panier, c'est ce bloc de code, présent dans le fichier *basket.php*, qui est utilisé :

```
if (isset($_GET['del']) && isset($_GET['del1'])) {    $dbConnection->del($_GET['del'],$_GET['del1']);
}
```

Lorsque l'utilisateur clique sur le lien de suppression suivant (il est donc connecté) le test ci-dessus est vrai, c'est donc la méthode *del()* de l'objet *\$dbConnection* qui sera utilisée :

```
<div class="product-delete">
  <a href="basket.php?del=<?=$product['productName'];?>&del1=<?=$product['fk_idUsers'];?>"> <h3>X</h3> </a>
</div>
```

Cette méthode permettra la suppression de l'article dont les informations auront été passée en arguments (voir figure suivante).

```
public function del($productName,$fk_idUser) {

    $db = $this->_dbConnection;
    $tmpRes = $db->prepare("DELETE FROM Baskets WHERE productName = :productName AND fk_idUsers = :fk_idUser");

    $tmpRes->bindValue(":productName",$productName);
    $tmpRes->bindValue(":fk_idUser",$fk_idUser);

    $tmpRes->execute();

    $tmpRes->closeCursor();
}
```

Dans l'autre cas (utilisateur non-connecté), c'est ce bloc de code qui sera utilisé :

```
if (isset($_GET['del2'])) {    $basket->del($_GET['del2']);
}
```

... après avoir cliqué sur le lien de suppression suivant :

```
<div class="product-delete">
  <a href="basket.php?del2=<?=$product['idProducts'];?>"> <h3>X</h3> </a>
</div>
```

C'est alors la méthode del() de l'objet \$basket qui sera utilisée pour supprimer l'article :

```
public function del($product_id) {
    unset($_SESSION['basket'][$product_id]);
}
```

IV- Test du système mis en place

4a. Test du système d'ajout au panier (utilisateur connecté)

L'utilisateur John Doe est connecté et il ajoute plusieurs produits à son panier :

Serious

Petit vélo de ville, très pratique pour les petits trajets.

Prix : 1 200,00€

Quantité : 2

Rocket

Le vélo électrique le plus rapide jamais fabriqué.

Prix : 1 199,99€

Quantité : 1

Total HT 2 399,99€

Total TTC 2 879,99€

Page produits

Les articles sont donc ajoutés à la base de données, dans la table **Baskets** :

<div><div><div>←</div><div>→</div></div><div></div></div>				idBaskets	productName	description	quantity	price	fk_idUsers
<div><div></div><div>✎ Éditer</div><div>📄 Copier</div><div>✖ Supprimer</div></div>	2	Rocket	Le vélo électrique le plus rapide jamais fabriqué.	1	1199.99	4			
<div><div></div><div>✎ Éditer</div><div>📄 Copier</div><div>✖ Supprimer</div></div>	4	Serious	Petit vélo de ville, très pratique pour les petits...	2	600	1			
<div><div></div><div>✎ Éditer</div><div>📄 Copier</div><div>✖ Supprimer</div></div>	5	Rocket	Le vélo électrique le plus rapide jamais fabriqué.	1	1199.99	1			

Si l'utilisateur supprime les articles, ils disparaissent de la base de données.

Après déconnexion et reconnexion, les articles de l'utilisateur seront toujours présents dans le panier.

4b. Test du système d'ajout au panier (utilisateur non-connecté)

Un utilisateur non-connecté ajoute des produits à son panier :



Serious
Petit vélo de ville, très pratique pour les petits trajets.
Quantité : 1
Prix : 600,00€



Rocket
Le vélo électrique le plus rapide jamais fabriqué.
Quantité : 1
Prix : 1 199,99€

Total HT: 1 799,99€
Total TTC: 2 159,99€

[Page produits](#)

Les articles ne sont donc pas stockés en base de données mais sont stockés en session.

Dès la fermeture du navigateur, les produits sont supprimés de la session après destruction de la session.

Conclusion

En conclusion, on peut dire que la mise en place d'un système de panier est impérative dans le cadre du développement d'un site de e-commerce. En effet, la mise en place d'un tel système permettra aux utilisateurs de stocker leurs articles le temps de la visite du site, jusqu'au paiement des articles, à l'image d'un panier dans un supermarché.

Pour permettre le déploiement d'un système de panier dans ce type de projet, il est nécessaire de mettre en place un certain nombre de module, tel que :

- Un système de panier stocké en session, avec l'utilisation d'un système d'espaces personnels déjà en place ;
- Un système de panier stocké en session, donc via un système qui aura été implémenté avant ou pendant la réalisation de ce système panier.

Le tout fonctionnant de manière homogène et permettant d'améliorer la satisfaction des utilisateurs.