

ARGUELLES Alexian  
GADBIN Hugo  
DELAPLAGNE Titouan  
ABAYEV Amina  
Grp 1B4

## **Documentation PHP**

Semaine 50 : Code et trace d'exécution PHP pour les  
User Stories prioritaires



<b>Introduction.....</b>	<b>3</b>
<b>1. Créer un compte utilisateur.....</b>	<b>4</b>
User Story.....	4
Résultat.....	4
Code de traitement de la page du formulaire.....	5
Trace d'exécution.....	6
<b>2. Se connecter.....</b>	<b>8</b>
User Story.....	8
Résultat.....	8
Code de traitement du formulaire de connexion :.....	9
Trace d'exécution.....	10
<b>3. Consulter un produit.....</b>	<b>12</b>
User Story.....	12
Code.....	12
Trace d'exécution.....	17
<b>4. Voir le stock disponible.....</b>	<b>18</b>
User Story.....	18
Code.....	18
Trace d'exécution.....	20
<b>5. Consulter sans acheter produit en rupture.....</b>	<b>20</b>
User Story.....	20
Code.....	20
Trace d'exécution.....	21
<b>6. Naviguer facilement par un menu.....</b>	<b>22</b>
User Story.....	22
Code.....	23
Trace d'exécution.....	26
<b>7. Ajouter un nouvel article.....</b>	<b>27</b>
User story :.....	27
Code :.....	27
Trace d'exécution :.....	31
<b>8. Consulter la liste des produits.....</b>	<b>35</b>
User story :.....	35
Code :.....	35
Trace d'exécution :.....	37
<b>9. Trier par catégorie.....</b>	<b>38</b>
Code :.....	38
Trace d'exécution :.....	39
<b>10. Filtrer par prix.....</b>	<b>40</b>
Code :.....	40

# Introduction

Dans le cadre du développement de l'application E-Commerce, les **User Stories prioritaires** définissent les fonctionnalités essentielles à implémenter en premier.

Dans ce compte rendu, nous présentons le **code PHP** correspondant à chacune de ces fonctionnalités, ainsi que la **trace d'exécution**.

# 1. Créer un compte utilisateur

## User Story

### User Story

En tant que client, je veux pouvoir créer un compte utilisateur afin de pouvoir gérer mes informations personnelles, suivre mes commandes et bénéficier d'une expérience personnalisée.

### Importance

Must (Default)

### Critères d'acceptation

- Accès à la création de compte : Le client doit pouvoir accéder facilement à la page d'inscription via un bouton visible, accessible depuis toutes les pages du site. ...
- Formulaire d'inscription : Le formulaire doit inclure des champs pour l'email, le mot de passe (avec critères de sécurité), et une confirmation du mot de passe. ...
- Accès à l'espace utilisateur : Une fois inscrit, le client doit pouvoir se connecter à son espace personnel pour gérer ses informations, suivre ses commandes, et consulter son historique d'achats. ...

## Résultat

[Lien vers le formulaire de création de compte](#)

**Créer un compte**

<input type="text" value="Prénom"/>		
<input type="text" value="Nom"/>		
<input type="text" value="Genre"/>		
<b>Date de naissance</b>		
Jour	Mois	Année
<input type="text" value="nom@example.com"/>		
<input type="text" value="Numéro de téléphone"/>		
<input type="text" value="Mot de passe"/>		
<input type="text" value="Confirmer le mot de passe"/>		
<b>Adresse</b>		
<input type="text" value="Adresse"/>		
<input type="text" value="Complément d'adresse"/>		
<input type="text" value="Code postal"/>		
<input type="text" value="Ville"/>		
<input type="text" value="Pays"/>		
<b>S'inscrire</b>		

## Code de traitement de la page du formulaire

```
<?php
    session_start();
    require_once '../components/bd.php';

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $email = htmlspecialchars($_POST['email']);

        $result = $pdo->prepare("SELECT email FROM Client WHERE email = :email");
        $result->execute([':email' => $email]);
        $row = $result->fetch();

        if ($row) {
            echo "Email déjà utilisé";
        } else {
            // Insertion dans la base de données
            $insert = $pdo->prepare("INSERT INTO Client (email, password) VALUES (:email, :password)");
            $insert->execute([':email' => $email, ':password' => $password]);
            echo "Inscription réussie";
        }
    }
}
```

## Trace d'execution

- Une fois les champs remplis, si tout est bon, ça nous ramène sur la page d'accueil, en étant connecté et les informations sont stockées dans la base de données.

54	doc-php@iut-blagnac.fr	client	\$2y\$10\$vslbi	Arguelles	Alexian	1111111111	H	2004-09-07	104
104	31700	Blagnac	Exemple adresse						

- Si une information qu'on à rempli ne convient pas, une erreur s'affiche sur le formulaire.

The screenshot shows a user interface for a form. At the top, there is a dropdown menu labeled "Genre". Below it, there is a field labeled "Date de naissance" with a dropdown arrow. A tooltip-like message box is overlaid on the dropdown arrow, containing an exclamation mark icon and the text "Sélectionnez un élément dans la liste." (Select an item in the list).

## 2. Se connecter

### User Story

**User Story**

En tant que client ayant déjà commandé sur le site je souhaite pouvoir me connecter à mon compte afin d'avoir accès à ce dernier

**Importance**

Must

**Critères d'acceptation**

- Page de connexion disponible ...
- Permettre l'accès aux données propres à mon compte ...

### Résultat

[Lien vers le formulaire de connexion](#)

### Connexion

E-mail ou numéro de téléphone

Mot de passe

Se souvenir de moi

**Se connecter**

Pas encore de compte ? [S'inscrire](#)

**Accéder au dashboard**

## Code de traitement du formulaire de connexion :

```
<?php
    session_start();
    require_once '../components/bd.php';

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $email = htmlspecialchars($_POST['email']);
        $password = htmlspecialchars($_POST['password']);

        // Récupérer l'utilisateur avec son email depuis la base de données
        $result = $pdo->prepare("SELECT idClient, email, password, prenom FROM Client
WHERE email = :email");
        $result->execute(['email' => $email]);

        $client = $result->fetch();

        if ($client) {

            if (password_verify($password, $client['password'])) {

                echo "Connexion réussie !";

                if (isset($_POST['remember-me'])) {
                    setcookie('client_email', $email, time() + 60 * 60 * 24 * 7);
                } else {
                    setcookie('client_email', $email, time() - 1);
                }

                $_SESSION["idClient"]=$client["idClient"];
                $_SESSION['client_email'] = $email;
                $_SESSION['client_prenom'] = $client['prenom'];

                header("Location: ../index.php");
            } else {

                echo "Mot de passe incorrect.";
                header("Location: ../login.php?error=30");
            }
        } else {

            echo "Aucun utilisateur trouvé avec cet email.";
            header("Location: ../login.php?error=40");
        }
    }
?>
```

## Trace d'exécution

1. Les identifiant ne respecte pas le format :

The screenshot shows a 'Connexion' (Login) form. In the email input field, the user has typed 'qsdqsd'. A red error message box appears below the input field, stating: 'Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "qsdqsd".' (Please include "@" in the email address. There is no "@" symbol in "qsdqsd"). Below the input fields are two buttons: 'Se souvenir de moi' (Remember me) with an unchecked checkbox, and a large blue 'Se connecter' (Connect) button. At the bottom, there is a link 'Pas encore de compte ? S'inscrire' (Not registered? Sign up).

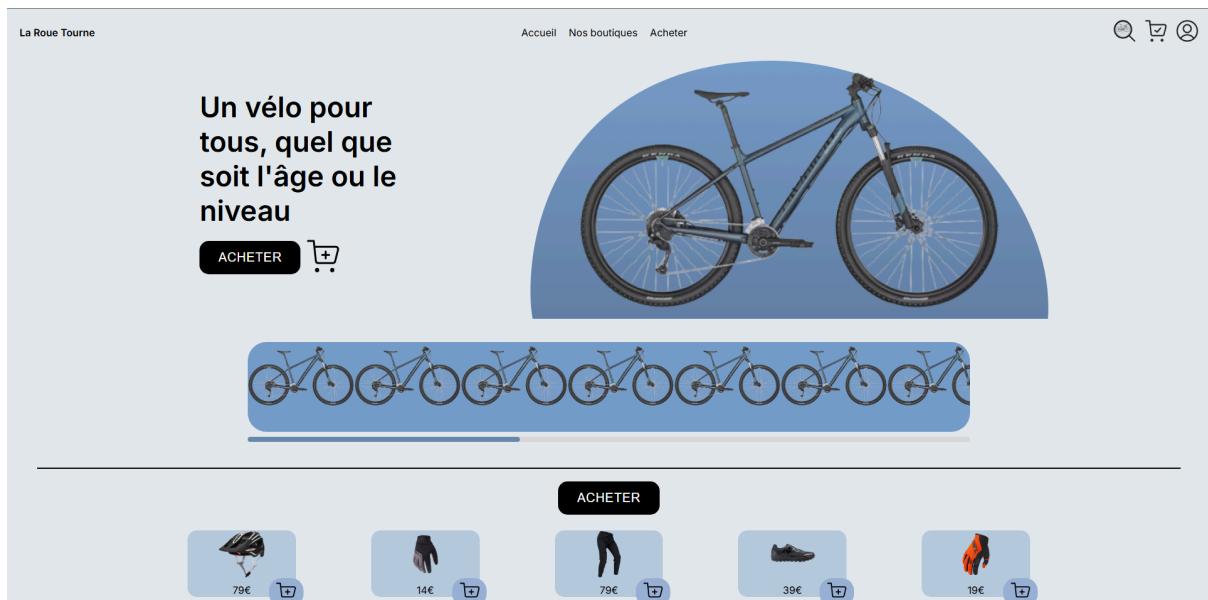
2. Le compte n'existe pas :

193.54.227.208/~R2024SAE3011/login.php?error=40

The screenshot shows a 'Connexion' (Login) form. It features two input fields: 'E-mail ou numéro de téléphone' (Email or phone number) and 'Mot de passe' (Password). Below the inputs are two buttons: an unchecked 'Se souvenir de moi' (Remember me) checkbox and a large blue 'Se connecter' (Connect) button. At the bottom, there is a link 'Pas encore de compte ? S'inscrire' (Not registered? Sign up).

Il y a un message d'erreur dans l'url, le formulaire se réinitialise.

3. La connexion est réussi, redirection vers la page d'accueil



### 3. Consulter un produit

#### User Story

En tant que client, je veux pouvoir consulter la description des produits, afin d'accéder à ces caractéristiques et son prix

##### Critères d'acceptation

- Accès au prix: pouvoir consulter le prix du produit
- Accès à la description: pouvoir consulter une description du produit

#### Code

Le code se découpe en trois parties,

La première est l'en-tête php. On y trouve l'inclusion du header.php

```
include './components/header.php';
```

Ainsi que l'inclusion de bd.php avec toutes les requêtes utiles dans la page. Il y a quatres requêtes et un appel à la procédure getAvgAvis.

```
require_once './components/bd.php';
```

La première requête récupère les informations du produit à afficher

```
$req = $pdo->prepare("SELECT * FROM Produit WHERE idProduit=:idProd");
$req->execute(["idProd" => $_GET["idProduit"]]);

$prod = $req->fetch();
```

La seconde récupère les couleurs associées.

```
$req = $pdo->prepare("SELECT DISTINCT Couleur_idCouleur, nomCouleur FROM
VariantArticle, Couleur WHERE Produit_idProduit=:idProd AND
Couleur_idCouleur=idCouleur");
$req->execute(["idProd" => $_GET["idProduit"]]);

$colors = $req->fetchAll();
```

La troisième, les tailles associées.

```
$req = $pdo->prepare("SELECT DISTINCT Taille_idTaille, nomTaille FROM
VariantArticle, Taille WHERE Produit_idProduit=:idProd AND
Taille_idTaille=idTaille");
$req->execute(["idProd" => $_GET['idProduit']]);

$tailles = $req->fetchAll();
```

La quatrième récupère toutes les combinaisons article taille couleur.

```
$req = $pdo->prepare("SELECT * FROM VariantArticle WHERE
Produit_idProduit=:idProd");
$req->execute(["idProd" => $_GET["idProduit"]]);
$variants = $req->fetchAll();
```

La Procédure getAvgAvis est appellée qui permet d'obtenir la moyenne des avis des produits et leur nombre

```
// on prépare la requête
$statement = $pdo->prepare('CALL getAvgAvis(:idProd, @avg, @count)');
// on donne la valeur de $pIdGrp au paramètre :idGrp de la requête préparée, en
// précisant que c'est de l'INTEGER.
$statement->bindParam(':idProd', $_GET["idProduit"], PDO::PARAM_INT);
// on exécute la requête, donc la procédure stockée
$statement->execute();
// il faut fermer le cursor sinon on ne peut pas récupérer la valeur renournée par
// la procédure stockée
$statement->closeCursor();

// on prépare la requête de récupération du paramètre OUT de la procédure stockée
$resultat = $pdo->prepare("select @avg as avg, @count as count");
// on execute
$resultat->execute();
// on récupère le résultat de cette requête
$avisProd = $resultat->fetch();
```

On arrondi ensuite la moyenne au .5 le plus proche

```
$avisProd["avg"] = round($avisProd["avg"] * 2) / 2;
```

On a ensuite une section plus html du code même si ce dernier est fusionné au php.

```
<div class="stockAlert">
    <h1 id="rupture">Rupture de Stock!!</h1>
    <h1 id="stockBas">Stock très faible! Moins de 5 exemplaires restants!</h1>
</div>
<div class="main">
    <div class="infoSpace">
        <div class="img" id="img">
            
        
        <br><br>
        <div id="stockDisplay">Stock disponible : --</div>
    </form>
</div>
</div>
</div>

```

Les sections d'intérêts sont, l'affichage du nombre d'étoiles du produit, où sont ajoutées cinq étoiles pleines, demi-pleines, ou vides.

```

<?php
    for ($i = 0; $i < floor($avisProd["avg"]); $i++) {
    ?>
        
    <?php
        foreach ($colors as $color) {
    ?>
        <option value="= $color["Couleur_idCouleur"] ?&gt;"&gt;&lt;?=
$color["nomCouleur"] ?&gt;&lt;/option&gt;
        &lt;?php
    }
    ?&gt;
&lt;/select&gt;
&lt;select name="taille" id="taille" onChange='updateStock()'&gt;

    &lt;?php
        foreach ($tailles as $taille) {
    ?&gt;
        &lt;option value="<?= $taille["Taille_idTaille"] ?&gt;"&gt;&lt;?=
$taille["nomTaille"] ?&gt;&lt;/option&gt;
        &lt;?php
    }
    ?&gt;
</pre

```

```
</select>
```

On peut aussi remarquer des champs cachés notamment celui transmettant l'id du produit à ajouter au panier.

```
<input type="hidden" name="idProd" value="=$GET['idProduit']?" />
```

La dernière section est la section JavaScript dédiée à la gestion du stock(cf. [Voir le stock disponible](#))

Lorsque l'on décide d'ajouter le produit au panier le formulaire est envoyé à la page traitement\_ajoutPanier.php où il est traité selon deux cas, selon si oui ou non l'utilisateur est connecté à son compte.

Si il est connecté, l'article et ces informations sont ajoutés à la base de données

```
if(isset($_SESSION["idClient"])) {  
  
    require_once "components/bd.php";  
  
    $req=$pdo->prepare("INSERT INTO Panier(Client_id, quantite,  
ProduitTailleCouleur_Produit_idProduit,  
ProduitTailleCouleur_Couleur_idCouleur,  
ProduitTailleCouleur_Taille_idTaille)  
                      VALUES (:idClient, :qte, :idProd,  
:idCouleur, :idTaille)");  
  
    $req->execute(["idClient" => $_SESSION["idClient"], "qte"=>  
$_GET["qte"], "idProd" => $_GET["idProd"], "idCouleur"=>  
$_GET["couleur"], "idTaille" => $_GET["taille"]]);  
  
}  
Si il ne l'est pas ces derniers sont ajoutés à une variable session,
```

```
else{  
  
    $newRow=true;  
  
    foreach($_SESSION["panierTemporaire"] as &$panier) {  
        if($panier["produit"]==$_GET["idProd"] &&  
$panier["couleur"]==$_GET["couleur"] &&  
$panier["taille"]==$_GET["taille"]) {  
            $panier["qte"]+=$_GET["qte"];  
            $newRow=false;  
        }  
    }  
}
```

```

if ($newRow)
    $_SESSION["panierTemporaire"] [] =[

        "produit" => $_GET["idProd"],
        "qte" => $_GET["qte"],
        "couleur" => $_GET["couleur"],
        "taille" => $_GET["taille"]

    ] ;

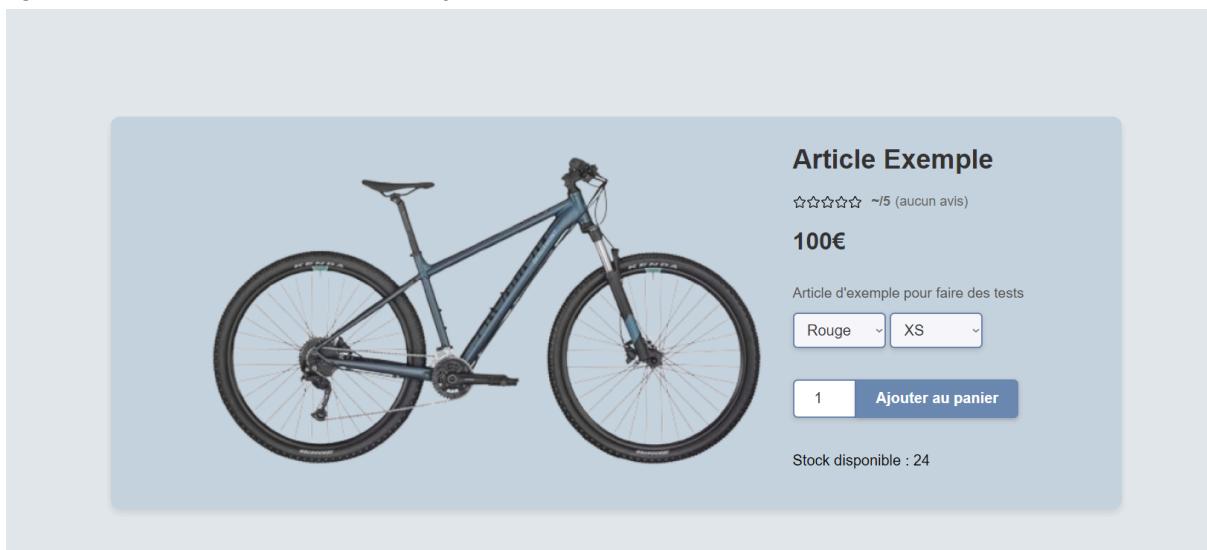
}

```

L'idée est qu'un client est probablement plus enclin à créer un compte si il a déjà fait son panier complet(il ne souhaiterait pas le perdre). Que si dès l'ajout du premier article on le force à créer un compte. Bien sûr, à la création et la connexion le panier sera transféré vers la base de données.

## Trace d'exécution

La page peut-être accédée de nombreuses façons, la seule condition est qu'elle soit appelée avec idProduit définit dans l'url sinon une redirection automatique est appliquée. A l'accès (idProduit=68), la page ressemble à ça, on peut y voir le nom de l'article, sa description, sa note, son prix, les couleurs et tailles pour lesquelles il est disponible, on peut également choisir la quantité et l'ajouter au panier.



## 4. Voir le stock disponible

### User Story

En tant que client, je souhaite voir la quantité de stock disponible pour chaque produit en temps réel afin de savoir si l'article est disponible avant de finaliser mon achat.

Critères d'acceptation:

- Affichage du stock disponible : La quantité de stock disponible doit être clairement affichée sur la page produit
- Indication de faible stock: Si le stock est inférieur à un seuil défini, un message d'alerte doit apparaître pour avertir l'utilisateur, comme "Stock très faible! Moins de 5 exemplaires restants!".

### Code

Pour cette User Story, l'indication de faible stock se trouve en haut de la page,

```
<h1 id="stockBas">Stock très faible! Moins de 5 exemplaires  
restants!</h1>
```

mais ne sera affiché que si le stock est bas, Le stock est affiché

```
<div id="stockDisplay">Stock disponible : --</div>
```

Mais est réglé par la méthode

```
function updateStock()
```

Celle-ci est appelée à chaque mise à jour des champs de sélection de couleur et de taille.

```
<select name="couleur" id="color" onChange="updateStock()">  
...  
<select name="taille" id="taille" onChange='updateStock()'>
```

Voici le corps de la fonction:

```
function updateStock() {  
  
    var eTaille = document.getElementById("taille");  
    var taille = eTaille.options[eTaille.selectedIndex].value;  
  
    var eCouleur = document.getElementById("color");  
    var couleur = eCouleur.options[eCouleur.selectedIndex].value;  
  
    // Parcourir le tableau des variantes pour trouver la correspondance  
    var stockTrouve = false; // Indicateur si on trouve le stock  
    for (var i = 0; i < variant.length; i++) {  
        if (  
            variant[i]["Couleur_idCouleur"] == couleur &&  
            variant[i]["Taille_idTaille"] == taille
```

```

    ) {
        // Exemple : mise à jour d'un champ HTML
        document.getElementById("stockDisplay").innerText =
            "Stock disponible : " + variant[i]["stock"];

        if(variant[i]['stock']==0) {
            document.getElementById("isAvailable").value="non";
            document.getElementById("rupture").hidden=false;
            document.getElementById("stockBas").hidden=true;
        }else{
            if(variant[i]["stock"]<5)
                document.getElementById("stockBas").hidden=false;
            else
                document.getElementById("stockBas").hidden=true;

            document.getElementById("isAvailable").value="oui";
            document.getElementById("rupture").hidden=true;
        }
        stockTrouve = true;
        break;
    }
}

if (!stockTrouve) {
    document.getElementById("stockDisplay").innerText = "Stock non
disponible";
}
}

```

Cette fonction trouve le stock en fonction de la taille et la couleur et actualise le stock disponible affiché, également si le stock du produit est bas mais pas vide(pour les ruptures de stocks, cf. [Consulter sans acheter produit en rupture](#)) la fonction affiche en haut de la page “Stock très faible! Moins de 5 exemplaires restants!”.

## Trace d'exécution

Si l'on appelle la page sur un article ayant un stock faible on voit cet affichage,

Stock très faible! Moins de 5 exemplaires restants!

Article Exemple

☆☆☆☆☆ ~/5 (aucun avis)

100€

Article d'exemple pour faire des tests

Rouge XS

1 Ajouter au panier

Stock disponible : 4

on peut voir que le stock est annoncé comme faible en haut et que le stock disponible est quatre(en bas à droite).

## 5. Consulter sans acheter produit en rupture

### User Story

En tant que client, je veux pouvoir voir les produits en rupture de stock, mais être empêché de les commander afin de ne pas perdre de temps en tentant d'acheter des articles qui ne sont pas disponibles

Critères d'acceptation:

- Affichage clair d'un message indiquant la rupture de stock
- Impossibilité d'ajout au panier d'article en rupture de stock
- Accessibilité aux produits en rupture de stock : Le client doit pouvoir consulter les informations des produits en rupture de stock, y compris la description, les caractéristiques et les avis, mais sans possibilité de finaliser l'achat.

### Code

Il y a deux sections à la gestion des produits en rupture de stock,  
La première est dans la page consulter produit où l'on peut trouver le message affiché en cas de rupture de stock,

```

<h1 id="rupture">Rupture de Stock!!</h1>
et dans la fonction
function updateStock()
...
if(variant[i]['stock']==0) {

document.getElementById("isAvailable").value="non";

document.getElementById("rupture").hidden=false;

document.getElementById("stockBas").hidden=true;
}

```

Ou dans le cas où l'article est en rupture de stock, dans cette couleur et cette taille, le bandeau est affiché. Ici est également passé en champs invisible du formulaire d'ajout au panier que l'article est indisponible.

La rupture de stock est gérée également dans le fichier "traitement\_ajoutPanier.php".

```

if($_GET["isAvailable"] != "oui") {
    header("Location: ../../consultProd.php?idProduit=".
$_GET["idProd"] . "&msg=". urlencode("Produit à court de stock dans
cette taille et couleur!"));
    exit();
}

```

Si le produit n'est pas disponible l'utilisateur est redirigé avec un message d'erreur.

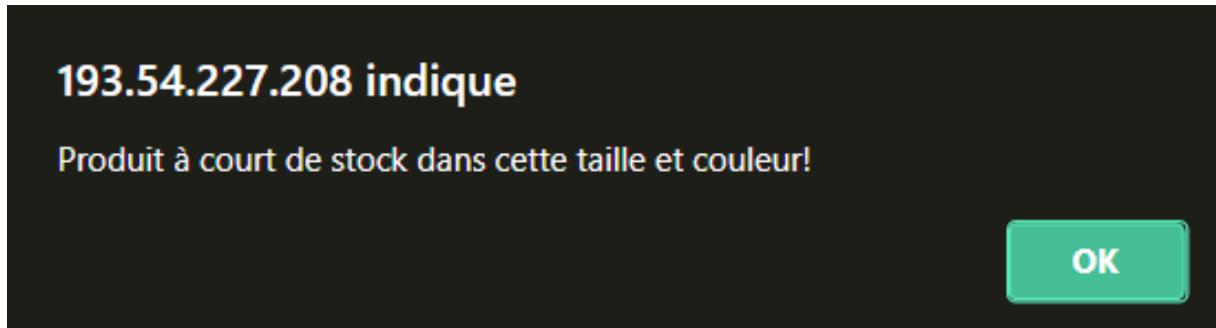
## Trace d'exécution

Si l'on consulte un produit en rupture de stock on arrive sur cette page,



On peut voir en grand en haut le message 'rupture de stock'.

À l'ajout au panier on est redirigé sur la page de consultation de l'article avec une alerte du navigateur,



Et nous revoici sur la même page

A screenshot of a product page for a bicycle. The page includes a header with "La Roue Tourne" and navigation links like Accueil, Nos boutiques, Acheter, and search icons. A red banner at the top says "Rupture de Stock!!". The main content shows a blue mountain bike and includes the following details:

- Article Exemple**
- Rating: ☆☆☆☆☆ ~5 (aucun avis)
- 100€**
- Description: Article d'exemple pour faire des tests
- Color: Rouge (dropdown menu)
- Size: XS (dropdown menu)
- Quantity: 1
- Action: **Ajouter au panier**
- Stock information: Stock disponible : 0

## 6. Naviguer facilement par un menu

### User Story

En tant que client je veux pouvoir naviguer facilement entre les pages et fonctionnalités à l'aide d'un menu présent sur toutes les pages

### Critères d'acceptation

- Avoir un fichier header.php permettant d'include facilement le menu requis.
- Afficher un menu ergonomique répondant aux maquettes du client

## Code

```
<?php
session_start();

$path = "./login.php";
if (isset($_GET["msg"])) {
    echo '<script>alert("' . $_GET["msg"] . '")</script>';
}

if (isset($_SESSION['client_email'])) {
    $path = "./profil.php";
}

require_once './components/bd.php';

$req = $pdo->prepare("SELECT * FROM Categorie WHERE idCategorieMere IS NULL");
$req->execute();

$meres = $req->fetchAll();

$req = $pdo->prepare("SELECT * FROM Categorie WHERE idCategorieMere IN (SELECT
idCategorie FROM Categorie WHERE idCategorieMere IS NULL)");
$req->execute();

$filles = $req->fetchAll();

?>
<!DOCTYPE html>
<html>

<head>
    <link
        href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap"
        rel="stylesheet">
        <link rel="stylesheet" href=".style/header/style.css">
        <link rel="stylesheet" href=".style/<?= basename($_SERVER["PHP_SELF"], ".php") ?>/style.css">
</head>

<header>

    <div class="header_title">
        <p>La Roue Tourné</p>
    </div>
    <div class="header_navigation">
        <a href="./index.php">
            <span>Accueil</span>
        </a>
```

```

<a href="./nosBoutiques.php">
    <span>Nos boutiques</span>
</a>
<div onmouseenter='document.getElementById("optionsAchat").hidden=false;' onmouseleave='document.getElementById("optionsAchat").hidden=true;'>
    <a href="./articles.php" >
        <span>Acheter</span>
    </a>
    <div id="optionsAchat" hidden>
        <div class="meres">
            <?php
                foreach ($meres as $mere) {
                    echo '<a href=("./articles.php?categories='.$mere["idCategorie"].' onmouseover="fillWith('.$mere["idCategorie"].' . ')">' . $mere["name"] . "</a>";
                }
            ?>
        </div>
        <div id="filles"></div>
    </div>
</div>

</div>
<div class="header_interaction">
    <a href="">
        
        </a>
    <a href="./panier.php" aria-label="Voir le panier">
        
        </a>
    <a href=<?php echo $path; ?>">
        
        </a>
</div>

<script>
    var filles = <?= json_encode($filles) ?>;
    function fillWith(idMere) {
        var htmlCode = '';
        filles.forEach(element => {
            if(idMere==element.idCategorieMere)
                htmlCode+= '<a href="./articles.php?categories=' + element.idCategorie +'"><p class="fille">' + element.name + "</p></a>";
        });
    }
</script>

```

```

        });

        document.getElementById('filles').innerHTML = htmlCode;
    }
</script>
</header>
```

Les sections intéressantes du menu sont,

```

if (isset($_GET["msg"])) {
    echo '<script>alert("' . $_GET["msg"] . '")</script>';
}
```

Cette section permet aux autres pages de se transmettre facilement des messages à afficher dans une alerte.

Ici on récupère les catégories mères de la base de données et leur catégories filles directes.

```

$req = $pdo->prepare("SELECT * FROM Categorie WHERE idCategorieMere IS NULL");
$req->execute();

$meres = $req->fetchAll();

$req = $pdo->prepare("SELECT * FROM Categorie WHERE idCategorieMere IN (SELECT
idCategorie FROM Categorie WHERE idCategorieMere IS NULL)");
$req->execute();

$filles = $req->fetchAll();
```

Ici on inclut le fichier “style.css” présent dans le dossier ayant le même nom que la page php actuel (si le header est inclus dans la page nosBoutiques.php, cette ligne définie comme css style/nosBoutiques/style.css

```
<link rel="stylesheet" href=".//style/<?= basename($_SERVER["PHP_SELF"], ".php")
?>/style.css">
```

On affiche ici les catégories mères quand Acheter est survolé dans le menu, on prépare également la place pour les catégories filles.

```

<div onmouseenter='document.getElementById("optionsAchat").hidden=false;' 
onmouseleave='document.getElementById("optionsAchat").hidden=true;'>
    <a href=".//articles.php" >
        <span>Acheter</span>
    </a>
    <div id="optionsAchat" hidden>
        <div class="meres">
            <?php
                foreach ($meres as $mere) {
                    echo '<a
href=.//articles.php?categories='.$mere["idCategorie"].' onmouseover="fillWith(' .
$mere["idCategorie"] . ')">' . $mere["name"] . "</a>";
                }
            ?>
```

```

        </div>
        <div id="filles"></div>
    </div>
</div>

```

Ici on a une fonction qui remplit la div #filles des catégories filles de la catégorie mère survolée

```

<script>
    var filles = <?= json_encode($filles) ?>

    function fillWith(idMere) {
        var htmlCode = '';

        filles.forEach(element => {
            if(idMere==element.idCategorieMere)
                htmlCode+= '<a href="#">'+ element.name + '</a>';
        });

        document.getElementById('filles').innerHTML = htmlCode;
    }
</script>

```

## Trace d'exécution

La Roue Tourne

Accueil Nos boutiques Acheter



On peut voir la barre à l'état passif et que lors du survol d' "Acheter", s'ouvre un menu avec les catégories mères.

Accueil Nos boutiques Acheter

Vélos Vélos électriques BMX Accessoires Composants

et lorsque l'on survole l'une d'elle,

Vélos Vélos électriques BMX Accessoires Composants

BMX Race

BMX Freestyle

BMX Street

BMX Park

les catégories filles associées.

On est si l'on appuie sur l'une des catégories ou sous catégories redirigée sur la page articles.php avec l'id de la categorie en paramètre, l'affichage des produits de cette dernière sera géré prochainement.

Lors de l'appui sur BMX Freestyle voilà l'url de redirection:

193.54.227.208/~R2024SAE3011/articles.php?categories=16

## 7. Ajouter un nouvel article

User story :

### User Story

En tant qu'administrateur, je souhaite pouvoir ajouter un nouvel article dans la catégorie VTT afin de mettre à jour facilement l'offre de produits disponibles dans cette catégorie spécifique du site

Code :

```

<div class="checkbox-container">
    <button type="button" id="showSizesBtn">Tailles</button>
    <div id="checkboxDropdownSizes" class="checkbox-dropdown" style="display: none;">
        <?php
            // Afficher les cases à cocher pour chaque taille
            foreach ($reqTailles->fetchAll() as $taille) {
                echo '<label><input type="checkbox" name="taille[]" value="" . htmlspecialchars($taille['idTaille'], ENT_QUOTES, "UTF-8") . "' . htmlspecialchars($taille['nomTaille'], ENT_QUOTES, "UTF-8") . '</label><br>';
            }
            $reqTailles->closeCursor();
        ?>
    </div>
</div>

<script>
    document.getElementById("showSizesBtn").addEventListener("click", function() {
        var checkboxDropdownSizes = document.getElementById("checkboxDropdownSizes");

        // Alterne l'affichage du conteneur
        if (checkboxDropdownSizes.style.display === "none") {
            checkboxDropdownSizes.style.display = "block";
        } else {
            checkboxDropdownSizes.style.display = "none";
        }
    });
</script>

```

```

<?php include './components/header.php';

if(isset($_POST["description"])){
    if (!empty($_FILES['imageProd']) AND $_FILES['imageProd']['error'] == 0) {
        // Testons si l'extension est autorisée
        $infosfichier = pathinfo($_FILES['imageProd']['name']);
        $extension_upload = $infosfichier['extension'];
        $extensions_autorisees = array('jpg');
        if (in_array($extension_upload, $extensions_autorisees) && 500000 > $_FILES["imageProd"]["size"]) {
            // On peut valider le fichier et le stocker définitivement
            move_uploaded_file($_FILES['imageProd']['tmp_name'], './assets/articlesImages/prod' . "1". ".jpg");
            header("Pragma: no-cache");
        }
        else {
            echo "Le fichier n'est pas du bon type ou il est trop volumineux !<br>";
        }
        // exit();
    }
}

include './components/bd.php';

// Récupérer les couleurs de la base de données
$reqCat = $pdo->prepare("SELECT * FROM Couleur");
$reqCat->execute();

// Récupérer les couleurs de la base de données
$reqTailles = $pdo->prepare("SELECT * FROM Taille");
$reqTailles->execute();

// Récupérer les catégories de la base de données
$reqCategories = $pdo->prepare("SELECT * FROM Categorie");
$reqCategories->execute();

?>

```

```

include './components/bd.php';

// Récupérer les couleurs de la base de données
$reqCat = $pdo->prepare("SELECT * FROM Couleur");
$reqCat->execute();

// Récupérer les tailles de la base de données
$reqTailles = $pdo->prepare("SELECT * FROM Taille");
$reqTailles->execute();

// Récupérer les catégories de la base de données
$reqCategories = $pdo->prepare("SELECT * FROM Categorie");
$reqCategories->execute();

?>

<body>
    <form class="main-container" action="#" method="POST">
        <div class="square-container">
            <div class="squareAdd">
                <!-- Bouton pour choisir une image -->
                <button type="button" id="chooseImageBtn">Choisir une image</button>

                <!-- Input pour sélectionner un fichier, masqué -->
                <input type="file" name="imageProd" id="uploadImage" accept="image/*" style="display: none;">

                <!-- Image de prévisualisation (masquée par défaut) -->
                <img id="previewImage" src="" alt="Prévisualisation de l'image" style="display: none;" class="preview-img">
            </div>
        </div>
    </form>
</body>

```

```

<script>
    // Références des éléments
    const chooseImageBtn = document.getElementById('chooseImageBtn');
    const uploadImage = document.getElementById('uploadImage');
    const previewImage = document.getElementById('previewImage');

    // Lorsqu'on clique sur le bouton, on déclenche l'input file
    chooseImageBtn.addEventListener('click', function() {
        uploadImage.click();
    });

    // Lorsque l'utilisateur sélectionne une image
    uploadImage.addEventListener('change', function(event) {
        const file = event.target.files[0];
        if (file) {
            const reader = new FileReader();

            // Affiche l'image sélectionnée dans le conteneur
            reader.onload = function(e) {
                previewImage.src = e.target.result; // Définit l'image comme source
                previewImage.style.display = 'block'; // Affiche l'image de prévisualisation

                // Déplace le bouton au-dessus de l'image
                chooseImageBtn.style.position = 'absolute';
                chooseImageBtn.style.top = '10px'; // Décalage du bouton au-dessus
                chooseImageBtn.style.zIndex = '20'; // Met le bouton au-dessus de l'image
            }
            reader.readAsDataURL(file);
        }
    });
</script>

```

```

</div>
<div class="form-container">
<div>
    <div class="checkbox-container">
        <button type="button" id="showCategoriesBtn">Catégories</button>
        <div id="checkboxDropdownCategories" class="checkbox-dropdown" style="display: none;">
            <?php
                // Afficher les cases à cocher pour chaque catégorie
                foreach ($reqCategories->fetchAll() as $categorie) {
                    echo '<label><input type="checkbox" name="categories[]" value="' . htmlspecialchars($categorie['idCategorie'], ENT_QUOTES, 'UTF-8') . '">' .
                        . htmlspecialchars($categorie['name'], ENT_QUOTES, 'UTF-8') . '</label><br>';
                }
                $reqCategories->closeCursor();
            ?>
        </div>
    </div>
</div>

<script>
    document.getElementById("showCategoriesBtn").addEventListener("click", function() {
        var checkboxDropdownCategories = document.getElementById("checkboxDropdownCategories");

        // Alterne l'affichage du contenu
        if (checkboxDropdownCategories.style.display === "none") {
            checkboxDropdownCategories.style.display = "block";
        } else {
            checkboxDropdownCategories.style.display = "none";
        }
    });
</script>

```

```

<input type="text" id="field3" name="prix" placeholder="Prix" required>
<input type="text" id="field4" name="nom" placeholder="Nom" required>

<div class="checkbox-container">
    <button type="button" id="showColorsBtn">Couleurs</button>
    <div id="checkboxDropdown" class="checkbox-dropdown" style="display: none;">
        <?php
            // Afficher les cases à cocher pour chaque couleur
            foreach ($reqCat->fetchAll() as $couleur) {
                echo '<label><input type="checkbox" name="couleur[]" value="' . htmlspecialchars($couleur['idCouleur'], ENT_QUOTES, 'UTF-8') . '">' .
                    . htmlspecialchars($couleur['nomCouleur'], ENT_QUOTES, 'UTF-8') . '</label><br>';
            }
            $reqCat->closeCursor();
        ?>
    </div>
</div>

<script>
    document.getElementById("showColorsBtn").addEventListener("click", function() {
        var checkboxDropdown = document.getElementById("checkboxDropdown");

        // Alterne l'affichage du contenu
        if (checkboxDropdown.style.display === "none") {
            checkboxDropdown.style.display = "block";
        } else {
            checkboxDropdown.style.display = "none";
        }
    });
</script>

```

```

<textarea id="description" name="description" placeholder="Description" required></textarea>

<input class="ajouter" type="submit" value="Ajouter"></input>

<script>

    // Fonction de validation
    function validateForm() {
        // Vérification des catégories
        const categoriesChecked = document.querySelectorAll('input[name="categories[]"]:checked');
        if (categoriesChecked.length === 0) {
            alert("Veuillez sélectionner au moins une catégorie.");
            return false;
        }

        // Vérification des couleurs
        const colorsChecked = document.querySelectorAll('input[name="couleur[]"]:checked');
        if (colorsChecked.length === 0) {
            alert("Veuillez sélectionner au moins une couleur.");
            return false;
        }

        // Vérification des tailles
        const sizesChecked = document.querySelectorAll('input[name="taille[]"]:checked');
        if (sizesChecked.length === 0) {
            alert("Veuillez sélectionner au moins une taille.");
            return false;
        }

        // Vérification de l'image
        const imageFile = document.getElementById("uploadImage").files.length;
        if (imageFile === 0) {
            alert("Veuillez télécharger une image.");
            return false;
        }

        // Si tout est validé, retourne true
        return true;
    }
}

</script>
</div>
</div>
</form>
</body>

</html>

```

Trace d'exécution :

The screenshot shows a user interface for creating a product. On the left, there is a large empty rectangular field with a dashed border, labeled "Choisir une image" (Select an image) at the bottom center. To the right of this field is a vertical column of input fields and buttons. At the top is a blue header bar labeled "Catégories". Below it is a white input field labeled "Prix" (Price). Another white input field below it is labeled "Nom" (Name). A blue header bar labeled "Couleurs" (Colors) follows. Another blue header bar labeled "Tailles" (Sizes) follows. Below these is a white input field labeled "Description". At the bottom of the column is a blue button labeled "Ajouter" (Add).

**Bouton “Choisir une image” :**

Ce bouton ouvre l'explorateur de fichiers pour sélectionner une image. Une fois l'image choisie, celle-ci s'affiche dans le cadre dédié.

**Bouton “Catégories” :**

Ce bouton ouvre un menu déroulant contenant des cases à cocher pour sélectionner une ou plusieurs catégories.



**Zone de texte “Prix” :**

Cette zone permet de saisir le prix du produit.

**Zone de texte “Nom” :**

Cette zone permet de renseigner le nom du produit.

**Bouton “Couleurs” :**

Ce bouton ouvre un menu déroulant contenant des cases à cocher pour choisir une ou plusieurs couleurs.



**Zone de texte “Description” :**

Cette zone permet de saisir une description détaillée du produit.

**Bouton “Tailles” :**

Ce bouton ouvre un menu déroulant contenant des cases à cocher pour sélectionner une ou plusieurs tailles.

**Tailles**

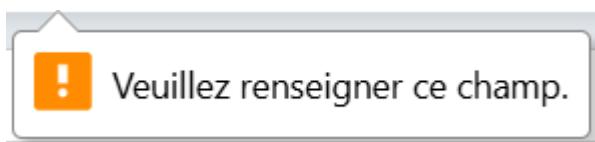
XS	<input type="checkbox"/>
S	<input type="checkbox"/>
M	<input type="checkbox"/>

**Bouton “Ajouter” :**

Ce bouton permet de valider et soumettre le formulaire, à condition que :

- tous les champs obligatoires sont remplis (nom, prix, description) ;
- au moins une catégorie soit sélectionnée ;
- au moins une couleur soit choisie ;
- au moins une taille soit cochée ;
- une image soit ajoutée.

Si une des conditions n'est pas remplie, la validation sera bloquée et un message d'erreur pourra s'afficher pour indiquer les champs ou sections manquants.



## 8. Consulter la liste des produits

User story :

### User Story

En tant que client je veux disposer d'une page dédiée où peuvent être trouvés tous les articles afin d'ensuite appliquer des filtres.

Code :

Récupération et Affichage des Articles :

```
$query = "
SELECT
    Produit.*,
    CASE
        WHEN Categorie.idCategorieMère IS NOT NULL THEN CatParent.name
        ELSE Categorie.name
    END AS categorie_name
FROM Produit
LEFT JOIN Categorie ON Produit.Categorie_idCategorie = Categorie.idCategorie
LEFT JOIN Categorie AS CatParent ON Categorie.idCategorieMère = CatParent.idCategorie
WHERE Produit.prix BETWEEN :prix_min AND :prix_max
";

$stmt = $pdo->prepare($query);
$stmt->bindValue(':prix_min', $prix_min, PDO::PARAM_INT);
$stmt->bindValue(':prix_max', $prix_max, PDO::PARAM_INT);
$stmt->execute();
$produits = $stmt->fetchAll();
```

Cette partie du code est responsable de récupérer tous les produits de la base de données, les organisant en fonction de leur catégorie. Cela assure que tous les articles sont disponibles pour être visualisés sur la page.

Pagination des Résultats :

```

$total_query = "
    SELECT COUNT(*)
    FROM Produit
    LEFT JOIN Categorie ON Produit.Categorie_idCategorie = Categorie.idCategorie
    WHERE Produit.prix BETWEEN :prix_min AND :prix_max
";

$total_stmt = $pdo->prepare($total_query);
$total_stmt->bindValue(':prix_min', $prix_min, PDO::PARAM_INT);
$total_stmt->bindValue(':prix_max', $prix_max, PDO::PARAM_INT);
$total_stmt->execute();
$total_articles = $total_stmt->fetchColumn();
$total_pages = ceil($total_articles / $articles_par_page);

```

Cette configuration permet de paginer les résultats pour que la page dédiée ne soit pas surchargée par un trop grand nombre de produits à la fois, facilitant la navigation et l'affichage.

## Construction et Affichage des Produits

```

<div class="grid">
    <?php foreach ($produits as $produit): ?>
        <div class="product-card">
            <?php
                $image = isset($images_categorie[$produit['categorie_name']])
                    ? $images_categorie[$produit['categorie_name']]
                    : 'assets/articles/default.png';
            ?>
            " class="product-image">
            <h4><?= htmlspecialchars($produit['nom']) ?></h4>
            <p class="price"><?= number_format($produit['prix'], 2, ',', ' ') ?> €</p>
            <p class="description"><?= htmlspecialchars($produit['description']) ?></p>
        </div>
    <?php endforeach; ?>
</div>

```

Chaque produit est affiché dans une grille, ce qui rend la page visuellement agréable et facile à naviguer. Les produits sont présentés avec leurs images, noms, descriptions, et prix, permettant aux utilisateurs de voir tous les articles disponibles d'un coup d'œil.

## Trace d'exécution :

La Roue Tourne

Accueil Nos boutiques Acheter

**Filtres**

**Catégories**

Accessoires pour téléphone et tablette  
 Antivols  
 Bidons et porte-bidons  
 Casques  
 Compteurs  
 Équipement de protection  
 Feux et radars  
 Garde-boue  
 Lunettes de vélo  
 Outils et entretien  
 Paniers  
 Peinture de retouche  
 Pompe  
 Porte-bagages pour vélo  
 Range vélo  
 Remorques  
 Sacs et sacoches  
 Sonnettes et avertisseurs

**BMX**

BMX Freestyle  
 BMX Park  
 BMX Race  
 BMX Street

**Composants**

Pneus  
 Roues

**Vélos**

**Numquam est** **361,77 €**  
Sunt ea vel at omnis nulla.

**Ratione accusantium** **492,40 €**  
Fugit blanditiis cum sapiente beatae.

**Reprehenderit aut** **475,68 €**  
Omnis et nisi voluptate voluptas natus.

**Consequuntur iste** **138,57 €**  
Nulla praesentium iusto molestiae querat tempore est consequatur eius.

**Quos id** **224,31 €**  
Pariatur voluptatem velit pariatur omnis et.

**Voluptatibus minus** **255,00 €**  
Eveniet soluta animi ut cumque doloremque sequi ab consequatur.

**Eveniet laboriosam** **36,62 €**  
Possimus repudiandae velit non sequi.

**Explicabo tempore** **204,58 €**  
Est id totam eos commodi suscipit.

**Consequatur porro** **44,28 €**  
Sequi quis nam et est quos.

Possibilité d'aller à la page suivante :

**Sed ab**  
**260,01 €**  
Non similique eos ullam qui adipisci maxime vel.

**Illum deleniti**  
**165,38 €**  
Ipsa quia itaque quia est quam neque quibusdam.

**Harum dolorum**  
**62,83 €**  
Non nisi eveniet dicta dolor ipsam natus.

1    2    Suivant »

## 9. Trier par catégorie

### User Story :

En tant que client, je veux pouvoir voir les vélos regroupés par catégories afin de trouver plus facilement le type de vélo qui correspond à mes besoins.

### Code :

Dans article.php :

Construction de la requête SQL avec les catégories :

```
if (!empty($categorie_ids)) {
    $placeholders = implode(',', array_map(fn($index) => ":cat_{$index}", array_keys($categorie_ids)));
    $query .= " AND (Produit.Categorie_idCategorie IN ($placeholders) OR Categorie.idCategorieMere IN ($placeholders));
```

Liaison des paramètres pour les catégories :

```
if (!empty($categorie_ids)) {
    foreach ($categorie_ids as $index => $id) {
        $stmt->bindValue(":cat_{$index}", $id, PDO::PARAM_INT);
    }
}
```

Récupération des catégories et affichage des filtres :

```
<?php
<?php foreach ($categories as $categorie): ?>
    <?php if (!$categorie['idCategorieMere']): ?>
        <li><strong><?= htmlspecialchars($categorie['name']) ?></strong></li>
        <!-- Sous-catégories -->
        <ul>
            <?php foreach ($categories as $subcat): ?>
                <?php if ($subcat['idCategorieMere'] == $categorie['idCategorie']): ?>
                    <li>
                        <input type="checkbox" name="categories[]" value="<?= $subcat['idCategorie'] ?>" <?= in_array($subcat['idCategorie'], $categorie_ids) ? 'checked' : '' ?>
                        <?= htmlspecialchars($subcat['name']) ?>
                    </li>
                <?php endif; ?>
            <?php endforeach; ?>
        </ul>
    <?php endif; ?>
<?php endforeach; ?>

```

## Trace d'exécution :

**Filtres**

**Catégories**

**Accessoires**

- Accessoires pour téléphone et tablette
- Antivols
- Bidons et porte-bidons
- Casques
- Compteurs
- Équipement de protection
- Feux et radars
- Garde-boue
- Lunettes de vélo
- Outils et entretien



**Officia iusto**  
225,92 €

Facilis laborum culpa natus qui praesentium.

Possibilité de choisir plusieurs catégories :

**Filtres**

**Catégories**

**Accessoires**

- Accessoires pour téléphone et tablette
- Antivols
- Bidons et porte-bidons
- Casques
- Compteurs
- Équipement de protection
- Feux et radars
- Garde-boue
- Lunettes de vélo
- Outils et entretien
- Paniers
- Peinture de retouche
- Pompe
- Porte-bagages pour vélo
- Range vélo
- Remorques
- Sacs et sacoches
- Sonnettes et avertisseurs

**BMX**

- BMX Freestyle
- BMX Park
- BMX Race
- BMX Street

**Composants**

- Pneus



**Numquam est**  
361,77 €

Sunt ea vel at omnis nulla.



**Molestiae ad**  
112,79 €

Hic quod inventore sunt eligendi ducimus.



**Officia iusto**  
225,92 €

Facilis laborum culpa natus qui praesentium.



**Officia nihil**  
54,71 €

Incidunt et illum eos itaque pariatur.



**Dolore aliquam**  
290,93 €

Sunt dolore numquam voluptas cumque ratione.



**Quae voluptatem**  
49,41 €

Vero repellat animi nulla quia pariatur voluptatem.



39

# 10. Filtrer par prix

## User Story :

En tant que client, je veux pouvoir filtrer les vélos selon une plage de prix afin de trouver des vélos qui correspondent à mon budget.

## Code :

Dans article.php :

Récupération des filtres de prix :

```
$prix_min = isset($_GET['prix_min']) ? (float)$_GET['prix_min'] : 0;
$prix_max = isset($_GET['prix_max']) ? (float)$_GET['prix_max'] : 10000;
```

Ces lignes récupèrent les valeurs minimum et maximum de prix soumises par l'utilisateur via le formulaire des filtres. Si aucun filtre n'est défini, les valeurs par défaut sont :

- 0 pour le prix minimum.
- 10000 pour le prix maximum .

Condition de prix dans la requête SQL :

```
$query = "
SELECT
    Produit.*,
    CASE
        WHEN Categorie.idCategorieMere IS NOT NULL THEN CatParent.name
        ELSE Categorie.name
    END AS categorie_name
FROM Produit
LEFT JOIN Categorie ON Produit.Categorie_idCategorie = Categorie.idCategorie
LEFT JOIN Categorie AS CatParent ON Categorie.idCategorieMere = CatParent.idCategorie
WHERE Produit.prix BETWEEN :prix_min AND :prix_max
";
```

Cette partie de la requête SQL applique une condition WHERE pour sélectionner uniquement les produits dont le prix est compris entre les valeurs :prix\_min et :prix\_max.

Champs de saisie des filtres de prix dans le formulaire :

```
<h4>Prix</h4>
<div class="price-range">
    <input type="number" name="prix_min" placeholder="Min (€)" value="<?= $prix_min ?>" min="0">
    <input type="number" name="prix_max" placeholder="Max (€)" value="<?= $prix_max ?>" min="0">
</div>
<button type="submit">Appliquer</button>
```

Ces champs dans le formulaire permettent à l'utilisateur de spécifier un prix minimum et un prix maximum. Les valeurs actuelles (\$prix\_min et \$prix\_max) sont affichées pour maintenir les filtres après une soumission.

## Trace d'exécution :

Peinture de retouche  
Pompes  
Porte-bagages pour vélo  
Range vélo  
Remorques  
Sacs et sacoches  
Sonnettes et avertisseurs

**BMX**

- BMX Freestyle
- BMX Park
- BMX Race
- BMX Street

**Composants**

- Pneus
- Roues

**Vélos**

- Gravel
- Vélo de route
- Vélo de ville
- Vélo pour enfant
- VTC
- VTT

**Vélos électriques**

- Gravel électrique
- Vélo de route électrique
- Vélo de ville électrique
- VTC électrique
- VTT électrique

**Prix**

0  50



Quae voluptatem  
**49,41 €**  
Vero repellat animi nulla quia pariatur  
voluptatem.



Inventore debitis  
**19,77 €**  
Omnis occaecati tempora nesciunt facilis  
qui.



Praesentium aut  
**11,25 €**  
Qui blanditiis sed suscipit sint debitis  
aliquid.

# Conclusion

En conclusion, ce compte rendu a permis de détailler le code PHP mis en œuvre pour les **User Stories prioritaires** du projet E-Commerce, ainsi que la trace d'exécution associée.