TYPES DE MAGASINS

INSERT INTO `shopstype`(`idshopstype`, `shopstype`)

VALUES

(1, 'epicerie'),

(2, 'bijouterie'),

(3, 'boulangerie'),

(4, 'pharmacie'),

(5, 'chocolaterie'),

(6, 'grandes surfaces') ;

UNITES

INSERT INTO `units`(`idunits`, `unitsname`)

VALUES

(1, 'gramme'),

(2, 'milligramme'),

(3, 'kg'),

(4, 'litre'),

(5, 'piece');

SHOPS

INSERT INTO `shops` (`idshops`, `shopsName`, `shopsstreetnb`, `shopsstreetname`, `shopspostal`, `shopscountry`, `shopsimg`, `shopslogo`, `shopstype\_idshopstype`)

VALUES ('1', 'Vival', '33', 'Rue Portalis', '83330', 'Le Beausset', 'vival.jpg', 'vivalLogo.png', '1');

INSERT INTO `shops` (`idshops`, `shopsName`, `shopsstreetnb`, `shopsstreetname`, `shopspostal`, `shopscountry`, `shopsimg`, `shopslogo`, `shopstype\_idshopstype`)

VALUES ('2', 'Saoya', '42', 'Avenue des Fleurs', '83330', 'Le Beausset', 'saoya.jpg', 'saoyaLogo.png', '2');

INSERT INTO `shops` (`idshops`, `shopsName`, `shopsstreetnb`, `shopsstreetname`, `shopspostal`, `shopscountry`, `shopsimg`, `shopslogo`, `shopstype\_idshopstype`)

VALUES ('3', 'Pharmacie des Arcades', '10', 'Avenue Victor Hugo', '83330', 'Le Beausset', 'pharmacieDesArcades.jpg', 'pharmacieArcadesLogo.png', '4');

INSERT INTO `shops` (`idshops`, `shopsName`, `shopsstreetnb`, `shopsstreetname`, `shopspostal`, `shopscountry`, `shopsimg`, `shopslogo`, `shopstype\_idshopstype`)

VALUES ('4', 'Carrefour', '123', 'Avenue du Commerce', '83330', 'Le Beausset', 'carrefour.jpg', 'carrefourLogo.png', '6');

INSERT INTO `shops` (`idshops`, `shopsName`, `shopsstreetnb`, `shopsstreetname`, `shopspostal`, `shopscountry`, `shopsimg`, `shopslogo`, `shopstype\_idshopstype`)

VALUES ('5', 'Casino', '12', 'Avenue du Pre', '83330', 'Le Beausset', 'casino.jpg', 'casinoLogo.png', '6');

RAYONS DES MAGASINS

INSERT INTO `department`(`iddepartment`, `departmentname`) VALUES ('1','bracelet');

INSERT INTO `department`(`iddepartment`, `departmentname`) VALUES ('2','fruits\_et\_legumes');

INSERT INTO `department`(`iddepartment`, `departmentname`) VALUES ('3','epicerie\_salee');

PRODUCTS

INSERT INTO `products`(`idproducts`, `productsname`, `productsdescription`, `productsprice`, `productsimg`, `productsweight`, `units\_idunits`, `department\_iddepartment`)

VALUES (1, 'Bracelet en or', 'Bracelet en or 18 carats avec un design élégant.', '499.99', 'braceletOr.jpg', '0.02', '1','1');

INSERT INTO `products`(`idproducts`, `productsname`, `productsdescription`, `productsprice`, `productsimg`, `productsweight`, `units\_idunits`, `department\_iddepartment`)

VALUES (2, 'Bracelet en argent', 'Bracelet en argent blanc', '199.99', 'braceletArgent.jpg', '0.02', '1','1');

INSERT INTO `products`(`idproducts`, `productsname`, `productsdescription`, `productsprice`, `productsimg`, `productsweight`, `units\_idunits`, `department\_iddepartment`) VALUES ('3','Chips Lays','Délicieuses chips de pomme de terre Lays', '2.50', 'chipsLays.jpg', '150', '1','3');

INSERT INTO `products`(`idproducts`, `productsname`, `productsdescription`, `productsprice`, `productsimg`, `productsweight`, `units\_idunits`, `department\_iddepartment`) VALUES ('4', 'Poireau', 'Frais et croquant, idéal pour vos recettes de soupes et de plats', '1.20', 'poireau.jpg', '200', '1','2');

INSERT INTO `products` (`idproducts`, `productsname`, `productsdescription`, `productsprice`, `productsimg`, `productsweight`, `units\_idunits`, `department\_iddepartment`)

VALUES ('5', 'Salade', 'Salade fraîche et croquante, parfaite pour accompagner vos plats', '2.50', 'salade.jpg', '150', '1', '2');

SHOPS HAS ROLE

INSERT INTO `shops\_has\_department`(`shops\_idshops`, `department\_iddepartment`, `products\_idproducts`) VALUES ('2','1','1')

INSERT INTO `shops\_has\_department`(`shops\_idshops`, `department\_iddepartment`, `products\_idproducts`) VALUES ('2','1','2')

INSERT INTO `shops\_has\_department`(`shops\_idshops`, `department\_iddepartment`, `products\_idproducts`) VALUES ('1','2','4')

Selection des produits par rayons (peu importe le magasin)

SELECT \* FROM products

JOIN department ON products.department\_iddepartment = department.iddepartment

WHERE department.iddepartment = 1

SELECT productsname,productsprice FROM products

JOIN department ON products.department\_iddepartment = department.iddepartment

JOIN shops\_has\_department ON shops\_has\_department.department\_iddepartment = department.iddepartment

JOIN shops ON shops\_has\_department.shops\_idshops = shops.idshops

WHERE shops\_has\_department.department\_iddepartment = 1 AND shops\_has\_department.shops\_idshops = 2;

Si vous utilisez un fichier **.env** pour stocker les informations de configuration de votre base de données, vous pouvez les récupérer dans votre script PHP en utilisant une bibliothèque comme PHP dotenv.

Voici comment vous pouvez procéder :

1. **Installer la bibliothèque PHP dotenv** : Tout d'abord, vous devez installer la bibliothèque PHP dotenv en utilisant Composer. Vous pouvez l'installer en exécutant la commande suivante dans votre terminal à la racine de votre projet :

bashCopy code

composer require vlucas/phpdotenv

1. **Utiliser PHP dotenv dans votre script PHP** : Ensuite, vous pouvez utiliser PHP dotenv pour charger les variables d'environnement à partir du fichier **.env** dans votre script PHP. Voici comment vous pouvez le faire :

phpCopy code

// Charger PHP dotenv require\_once \_\_DIR\_\_ . '/vendor/autoload.php'; // Charger les variables d'environnement à partir du fichier .env $dotenv = Dotenv\Dotenv::createImmutable(\_\_DIR\_\_); $dotenv->load(); // Connexion à la base de données en utilisant les variables d'environnement $dsn = 'mysql:host=' . $\_ENV['DB\_HOST'] . ';dbname=' . $\_ENV['DB\_DATABASE'] . ';port=' . $\_ENV['DB\_PORT']; $username = $\_ENV['DB\_USER']; $password = $\_ENV['DB\_PASSWORD']; try { $pdo = new PDO($dsn, $username, $password); // Activer les exceptions pour les erreurs PDO $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION); } catch (PDOException $e) { // En cas d'erreur de connexion, afficher l'erreur echo 'Erreur de connexion : ' . $e->getMessage(); exit(); } // Récupérer le mot de passe haché à partir de la base de données en utilisant l'email soumis $email = $\_POST['email']; $query = "SELECT password FROM user WHERE email = ?"; $stmt = $pdo->prepare($query); $stmt->execute([$email]); $hash = $stmt->fetchColumn();

Dans cet exemple, nous utilisons PHP dotenv pour charger les variables d'environnement à partir du fichier **.env**. Ensuite, nous utilisons ces variables pour établir une connexion à la base de données.

Haut du formulaire

Pour hasher des mots de passe en PHP, vous pouvez utiliser la fonction **password\_hash()**. Voici comment vous pouvez l'utiliser dans votre script pour hasher le mot de passe avant de le stocker dans la base de données :

phpCopy code

// Récupérer le mot de passe soumis $password = $\_POST['password']; // Hasher le mot de passe $hashedPassword = password\_hash($password, PASSWORD\_DEFAULT); // Enregistrer le mot de passe haché dans la base de données $query = "INSERT INTO user (email, password) VALUES (?, ?)"; $stmt = $pdo->prepare($query); $stmt->execute([$email, $hashedPassword]);

Dans ce code :

* **password\_hash()** prend deux paramètres : le mot de passe à hasher et l'algorithme de hachage à utiliser. **PASSWORD\_DEFAULT** utilise l'algorithme de hachage recommandé actuellement (actuellement Bcrypt), mais vous pouvez également spécifier un algorithme spécifique si nécessaire.
* Le mot de passe haché est ensuite inséré dans la base de données.

Lorsque vous vérifiez les mots de passe lors de la connexion (comme vous l'avez déjà fait avec **password\_verify()**), vous comparez le mot de passe soumis avec le mot de passe haché stocké dans la base de données. Voici un exemple :

phpCopy code

// Récupérer le mot de passe haché à partir de la base de données en utilisant l'email soumis $email = $\_POST['email']; $query = "SELECT password FROM user WHERE email = ?"; $stmt = $pdo->prepare($query); $stmt->execute([$email]); $hash = $stmt->fetchColumn(); // Vérifier si le mot de passe soumis correspond au mot de passe haché if (!$hash || !password\_verify($\_POST['password'], $hash)) { echo "Identifiants invalides."; }

function init() {

  let iddepartmentLinks = document.querySelectorAll("button");

  iddepartmentLinks.forEach(e => {

      e.addEventListener('click', () => {

        let iddepartment = e.value

        console.log(iddepartment)

          let url = `http://localhost:3003/get/products/${iddepartment}`;

          fetchProducts(url);

      });

  });

}

function fetchProducts(url) {

  fetch(url)

      .then(response => response.json())

      .then(data =>{

      const jsonData = JSON.stringify(data);

      const encodedData = encodeURIComponent(jsonData);

      window.location.href = `products?data=${encodedData}`;})

      .catch(error => console.error("Error:", error));

}

function displayProducts(products) {

  products.forEach(product => {

      let Title = document.querySelector('.card\_title');

      Title.textContent = product.productsname;

      let Shop = document.querySelector('.card\_shop');

      Shop.textContent = product.shopsName;

      let Weight = document.querySelector('.card\_weight');

      Weight.textContent= (`${product.productsweight} ${product.units\_idunits}`);

      let Description = document.querySelector('.card\_description');

      Description.textContent = product.productsdescription;

      let Price = document.querySelector('.card\_price');

      Price.textContent = (`${product.productsprice} ${product.euros}`);

  });

}

init();

<script>

  const params = new URLSearchParams(window.location.search);

        const jsonData = params.get("data");

        const data = JSON.parse(decodeURIComponent(jsonData));

        console.log(data);

        // Appel de la fonction pour afficher les produits

        displayProducts(data);

        function displayProducts(products) {

            products.forEach(product => {

                let Title = document.querySelector('.card\_title');

                Title.textContent = product.Productsname;

                let img = document.querySelector('#imageLink');

                img.src = `./public/img/products/${product.productsimg}`

                let shop = document.querySelector('.card\_shop');

                shop.textContent = product.shopsName;

                let weight = document.querySelector('.card\_weight');

                weight.textContent = `${product.productsweight} ${product.unitsname}`;

                let description = document.querySelector('.card\_description');

                description.textContent = product.productsdescription;

                let price = document.querySelector('.card\_price');

                price.textContent = `${product.productsprice} ${product.euros}`;

            });

        }

</script>

//PRODUCTS

import { displayProducts } from './products.js';

function init() {

  const params = new URLSearchParams(window.location.search);

  const jsonData = params.get("data");

  const data = JSON.parse(decodeURIComponent(jsonData));

  console.log(data);

  // Appel de la fonction pour afficher les produits

  displayProducts(data);

    let iddepartmentLinks = document.querySelectorAll("button");

    iddepartmentLinks.forEach(e => {

        e.addEventListener('click', () => {

            let iddepartment = e.value;

            console.log(iddepartment);

            let url = `http://localhost:3003/get/products/${iddepartment}`;

            fetchProducts(url);

        });

    });

}

function fetchProducts(url) {

    fetch(url)

        .then(response => response.json())

        .then(data => {

            displayProducts(data); // Utiliser la fonction displayProducts pour afficher les produits

            const jsonData = JSON.stringify(data);

            const encodedData = encodeURIComponent(jsonData);

            window.location.href = `products?data=${encodedData}`;

        })

        .catch(error => console.error("Error:", error));

}

init();

  const params = new URLSearchParams(window.location.search);

        const jsonData = params.get("data");

        const data = JSON.parse(decodeURIComponent(jsonData));

        console.log(data);

        // Appel de la fonction pour afficher les produits

        displayProducts(data);

        function displayProducts(products) {

            products.forEach(product => {

                let Title = document.querySelector('.card\_title');

                Title.textContent = product.Productsname;

                let img = document.querySelector('#imageLink');

                img.src = `./public/img/products/${product.productsimg}`

                let shop = document.querySelector('.card\_shop');

                shop.textContent = product.shopsName;

                let weight = document.querySelector('.card\_weight');

                weight.textContent = `${product.productsweight} ${product.unitsname}`;

                let description = document.querySelector('.card\_description');

                description.textContent = product.productsdescription;

                let price = document.querySelector('.card\_price');

                price.textContent = `${product.productsprice} ${product.euros}`;

            });

        }

<div class="cart">

<button class="cart-icon" onclick="toggleCart()">

<i class="fa fa-shopping-cart"></i>

</button>

<div class="cart-actions hidden">

<button class="delete-icon" onclick="deleteCartItem()">

<i class="fa fa-trash"></i>

</button>

<input type="number" class="item-quantity" value="1" min="1">

<button class="plus-icon" onclick="increaseQuantity()">

<i class="fa fa-plus"></i>

</button>

<button class="minus-icon hidden" onclick="decreaseQuantity()">

<i class="fa fa-minus"></i>

</button>

</div>

</div>