PRESENTATION PROJET NSI

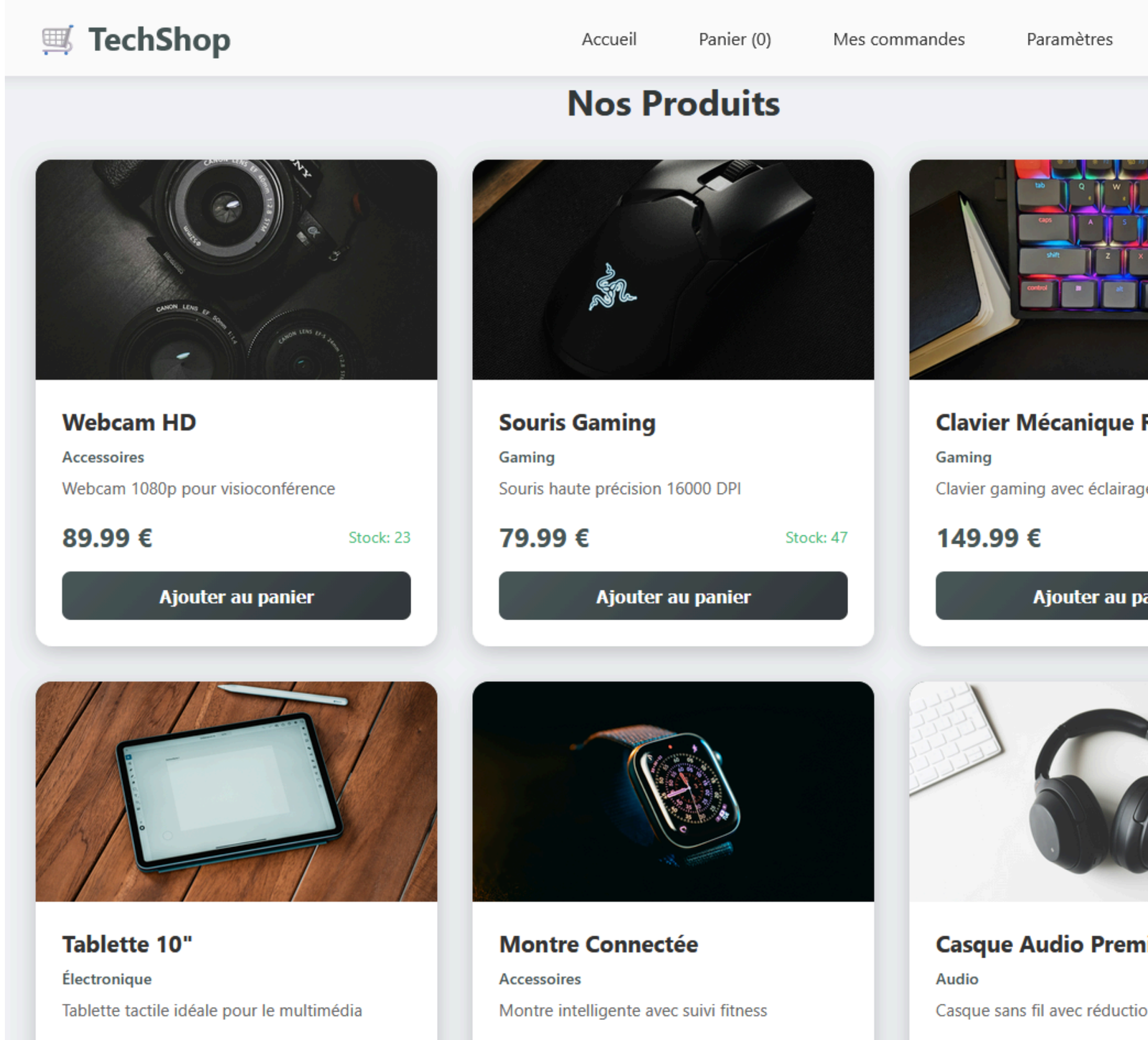
TECH-SHOP

AYOUB & MOHAND

CONTEXTE DU PROJET TECHSHOP

INTRODUCTION

TechShop est un site e-commerce complet développé en PHP et MySQL. Il permet aux utilisateurs de s'inscrire de manière sécurisée avec cryptage bcrypt, de parcourir un catalogue de produits, d'ajouter des articles à un panier dynamique géré par sessions PHP, et de passer commande avec un formulaire de paiement complet. Notre base de données comprend 5 tables relationnelles normalisées avec clés étrangères. Nous avons implémenté des requêtes préparées contre les injections SQL et des transactions pour garantir la cohérence des commandes. Le projet représente 16 fichiers PHP et 2500 lignes de code, avec une interface moderne et responsive. Nous sommes prêts à vous le démontrer en direct.



SOMMAIRE

PARTIE I

ARCHITECTURE DE LA BASE DE DONNÉES

SÉCURITÉ & CRYPTAGE

GESTION DES SESSIONS

PARTIE II

FONCTIONNALITÉS DU SITE

SYSTÈME DE PANIER DYNAMIQUE

PROCESSUS DE COMMANDE

INTRODUCTION AU

PROJET

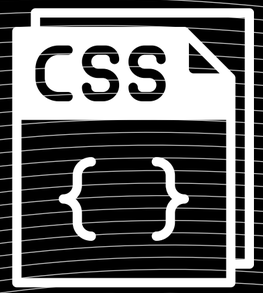
🎯 OBJECTIF DU PROJET

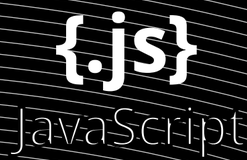
Créer un site e-commerce fonctionnel avec :

- ✅ Gestion d'utilisateurs sécurisée
- ✅ Catalogue de produits dynamique
- ✅ Panier d'achat persistant
- ✅ Système de paiement complet
- ✅ Historique des commandes

TECHNOLOGIES UTILISÉES

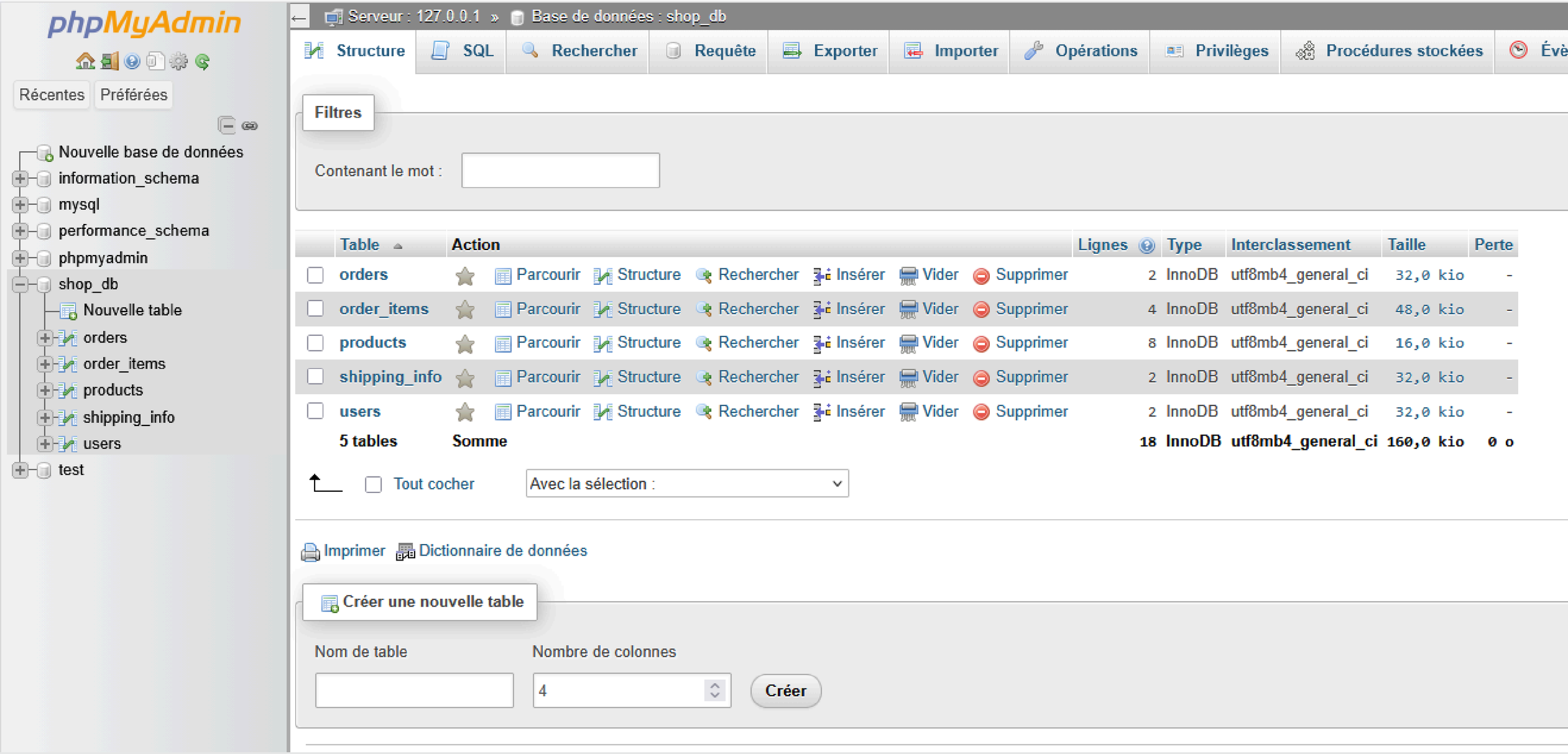
PHP 7+ | MySQL | HTML5 | CSS3 | JavaScript





ARCHITECTURE DE LA

BASE DE DONNÉES


phpMyAdmin
Récentes
Préférées
Nouvelle base de données
information_schema
mysql
performance_schema
phpmyadmin
shop_db
Nouvelle table
orders
order_items
products
shipping_info
users
test
Serveur : 127.0.0.1 » Base de données : shop_db
Structure
SQL
Rechercher
Requête
Exporter
Importer
Opérations
Privilèges
Procédures stockées
Filtres
Contenant le mot :
Table
Action
Lignes
Type
Interclassement
Taille
Perte
orders Parcourir Structure Rechercher Insérer Vider Supprimer 2 InnoDB utf8mb4_general_ci 32,0 kio -
order_items Parcourir Structure Rechercher Insérer Vider Supprimer 4 InnoDB utf8mb4_general_ci 48,0 kio -
products Parcourir Structure Rechercher Insérer Vider Supprimer 8 InnoDB utf8mb4_general_ci 16,0 kio -
shipping_info Parcourir Structure Rechercher Insérer Vider Supprimer 2 InnoDB utf8mb4_general_ci 32,0 kio -
users Parcourir Structure Rechercher Insérer Vider Supprimer 2 InnoDB utf8mb4_general_ci 32,0 kio -
5 tables Somme 18 InnoDB utf8mb4_general_ci 160,0 kio 0 o
Tout cocher
Avec la sélection :
Imprimer
Dictionnaire de données
Créer une nouvelle table
Nom de table
Nombre de colonnes
4
Créer

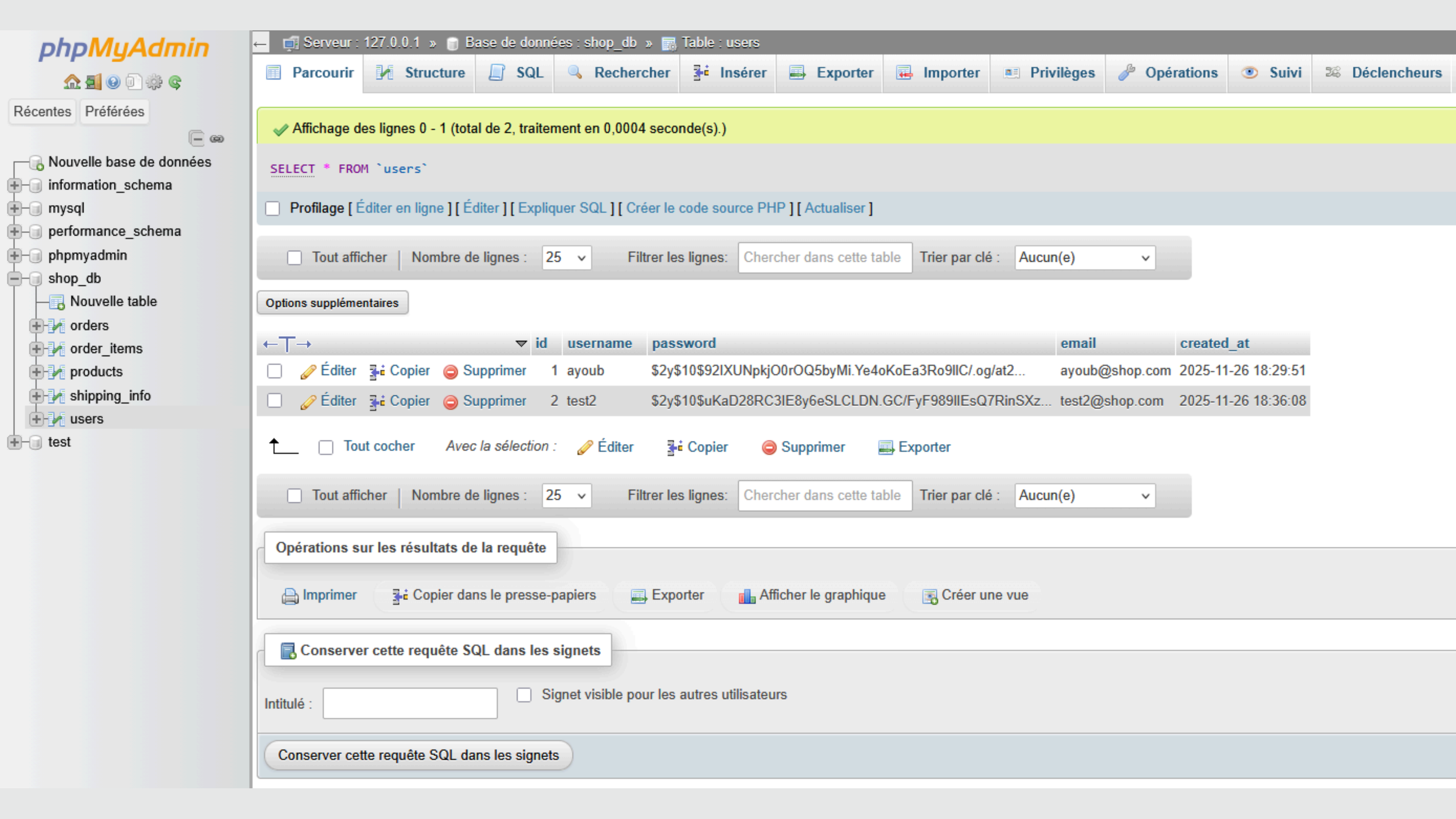
GESTION DES UTILISATEURS

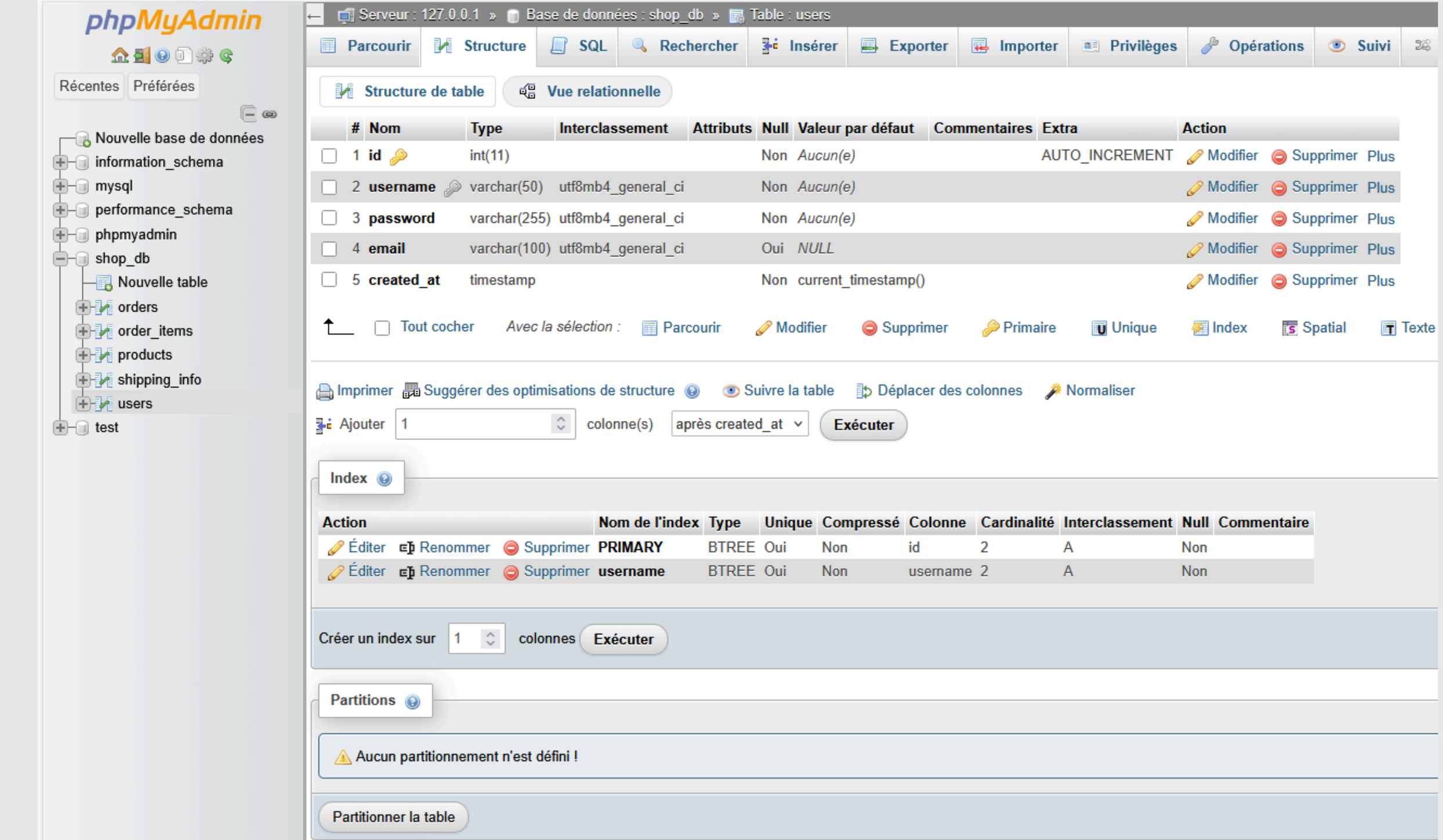
TABLE USERS

⚠️ PASSWORD stocke le HASH bcrypt (255 car)
🔒 Assure une Sécurité maximale

Presenté dans les prochaines slides

CHAMPS	*TYPE*	*CONTRAINTES*
id	INT	PRIMARY KEY
username	VARCHAR(50)	UNIQUE, NOT NULL
password	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	
created_at	TIMESTAMP	DEFAULT NOW


phpMyAdmin
Récentes
Préférées
Nouvelle base de données
information_schema
mysql
performance_schema
phpmyadmin
shop_db
Nouvelle table
orders
order_items
products
shipping_info
users
test
Serveur : 127.0.0.1 » Base de données : shop_db » Table : users
Parcourir
Structure
SQL
Rechercher
Insérer
Exporter
Importer
Privilèges
Opérations
Suivi
Déclencheurs
Affichage des lignes 0 - 1 (total de 2, traitement en 0,0004 seconde(s).)
SELECT * FROM `users`
Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]
Tout afficher
Nombre de lignes : 25
Filtrer les lignes: Chercher dans cette table
Trier par clé : Aucun(e)
Options supplémentaires
id username password email created_at
Éditer Copier Supprimer 1 ayoub $2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... ayoub@shop.com 2025-11-26 18:29:51
Éditer Copier Supprimer 2 test2 $2y$10$uKaD28RC3IE8y6eSLCLDN.GC/FyF989llEsQ7RinSXz... test2@shop.com 2025-11-26 18:36:08
Tout cocher
Avec la sélection :
Éditer
Copier
Supprimer
Exporter
Opérations sur les résultats de la requête
Imprimer
Copier dans le presse-papiers
Exporter
Afficher le graphique
Créer une vue
Conserver cette requête SQL dans les signets
Intitulé :
Signet visible pour les autres utilisateurs
Conserver cette requête SQL dans les signets


phpMyAdmin
Récentes
Préférées
Nouvelle base de données
information_schema
mysql
performance_schema
phpmyadmin
shop_db
Nouvelle table
orders
order_items
products
shipping_info
users
test
Serveur : 127.0.0.1 » Base de données : shop_db » Table : users
Parcourir
Structure
SQL
Rechercher
Insérer
Exporter
Importer
Privilèges
Opérations
Suivi
Structure de table
Vue relationnelle
Nom Type Interclassement Attributs Null Valeur par défaut Commentaires Extra Action
1 id int(11) Non Aucun(e) AUTO_INCREMENT Modifier Supprimer Plus
2 username varchar(50) utf8mb4_general_ci Non Aucun(e) Modifier Supprimer Plus
3 password varchar(255) utf8mb4_general_ci Non Aucun(e) Modifier Supprimer Plus
4 email varchar(100) utf8mb4_general_ci Oui NULL Modifier Supprimer Plus
5 created_at timestamp Non current_timestamp() Modifier Supprimer Plus
Tout cocher
Avec la sélection :
Parcourir
Modifier
Supprimer
Primaire
Unique
Index
Spatial
Texte
Imprimer
Suggérer des optimisations de structure
Suivre la table
Déplacer des colonnes
Normaliser
Ajouter 1 colonne(s) après created_at Exécuter
Index
Action Nom de l'index Type Unique Compressé Colonne Cardinalité Interclassement Null Commentaire
Éditer Renommer Supprimer PRIMARY BTREE Oui Non id 2 A Non
Éditer Renommer Supprimer username BTREE Oui Non username 2 A Non
Créer un index sur 1 colonnes Exécuter
Partitions
Aucun partitionnement n'est défini !
Partitionner la table

CRYPTAGE AVEC

BCRYPT

ALGORITHME BCRYPT :

Mot de passe : "ayb2008"

↓

password_hash($password, PASSWORD_DEFAULT)

↓

Hash : $2y$10$92IXUNpkjO0rOQ5byMi.Ye...

🔐 SÉCURITÉ : CRYPTAGE DES MOTS DE PASSE

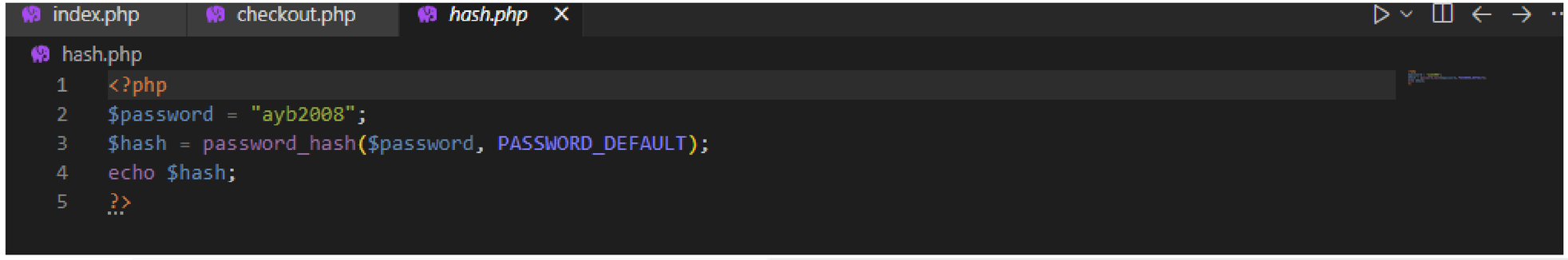
✅ Hachage unidirectionnel (impossible à décrypter)
✅ Salt automatique intégré
✅ Coût configurable (2^10 = 1024 itérations)
✅ Résistant aux attaques par force brute

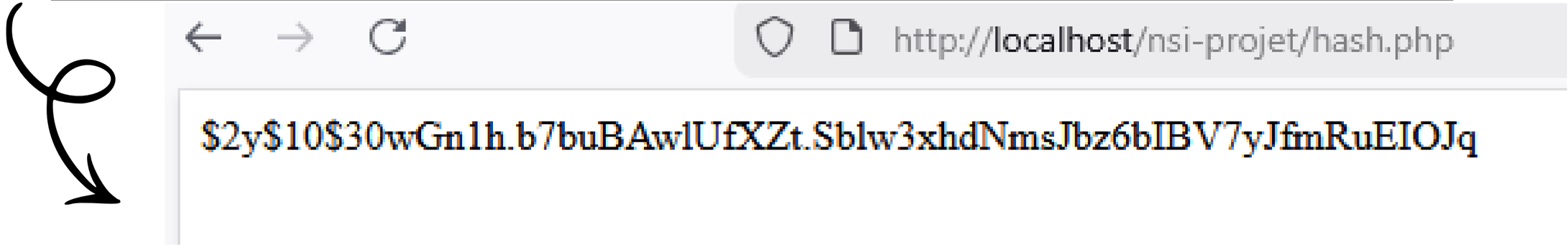
```
// Inscription
$hash = password_hash($password, PASSWORD_DEFAULT);

// Connexion
if (password_verify($password, $hash)) {
        // ✅ Authentification réussie
}
```

Test Bcrypt

Pour essayer de crypter un mot de passe j'ai fait un fichier php que vous pouvez utiliser ou vous entrer votre mot de passe ecrit puis vous lancer le code (en ouvrant votre localhost puis dans le fichier hash.php) pour tester, sinon les mot de passe se cryptent automatiquement quand l'utilisateur s'inscrit.

- Entrer le texte que vous voulez crypter dans $password = "votremdp";
- lancer xamp ou votre logiciel de serveur
- ouvrez : localhost/projet-nsi/hash.php





PROTECTION CONTRE LES

INJECTION SQL

🛡️ *PROTECTION CONTRE LES INJECTIONS SQL*

❌ *MAUVAISE PRATIQUE (Vulnérable)*

```
$query = "SELECT * FROM users WHERE username='$username'";
```

✅ *BONNE PRATIQUE (Sécurisée) → login.php ligne11&12*

```
$stmt = $conn->prepare("SELECT * FROM users WHERE username = ?");
$stmt->execute([$username]);
```

AVANTAGES :

- Séparation code SQL / données utilisateur
- Impossible d'injecter du SQL malveillant
- Protection automatique des caractères spéciaux

→ login.php ligne11&12

SYSTEME D'AUTHENTIFIQUATION

SYSTEME PHP

💾 Persistance pendant toute la session utilisateur
🔒 Données sécurisées côté serveur

CONNEXION
→ Création $_SESSION['user_id']
→ Stockage côté serveur

NAVIGATION
→ Vérification isLoggedIn()
→ Protection des pages sensibles

DÉCONNEXION
→ session_destroy()
→ Suppression totale

login.php

```php
<?php
require_once 'config.php';

$error = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'] ?? '';
    $password = $_POST['password'] ?? '';

    if ($username && $password) {
        $stmt = $conn->prepare("SELECT * FROM users WHERE username = ?");
        $stmt->execute([$username]);
        $user = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($user && password_verify($password, $user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];
            header('Location: index.php');
            exit();
        } else {
            $error = 'Identifiants incorrects';
        }
    } else {
        $error = 'Veuillez remplir tous les champs';
    }
}
?>
```

isLoggedIn() est utilise sur plusieurs pages pour vérifier si l'utilisateur est déjà connecte, s'il est connecte il le dirige vers l'accueil(1) sinon il dirige l'utilisateur vers la page de connexion/inscpriton(2)

1.)

```php
if (isLoggedIn()) {
    header('Location: index.php');
    exit();
}
```

2.)

```php
if (!isLoggedIn()) {
    header('Location: login.php');
    exit();
}
```

logout.php 1, M

```php
<?php
// fichier de deconnexion
session_start();
session_destroy();
header('Location: index.php');
exit();
?>
```

CLES ETRANGERS & INTEGRITÉ

RELATION ENTRE LES TABLES

EXEMPLE : Table ORDERS

```sql
CREATE TABLE orders (
    id INT PRIMARY KEY,
    user_id INT,
    total DECIMAL(10,2),
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

GARANTIT :

✅ user_id existe forcément dans users

❌ Impossible de créer une commande orpheline

🔄 Mises à jour en cascade possibles

BASE DE DONNEE

NORMALISÉ

AVANT : non normalisé

commande_id	user	produit	prix
1	jean	Laptop	1299
1	jean	souris	79

Aprés (Normalisé)

orders : 1 | user_id 1 | total 1378
order_items : 1 → Laptop, 1 → souris
users : 1 | jean

✅ Pas de redondance
✅ Cohérence garantie
✅ Facilité de mise à jour

FONCTIONNALITÉES

PRINCPALES

🏠 PAGE D'ACCUEIL

→ Catalogue de 8 produit
→ Filtrage par catégories
→ ajout au panier instantané

👤 GESTION DE COMPTE

→ Inscription avec validation
→ connexion sécurisé
→ modification du profil
→ changement de mot de passe

🛒 SYSTÈME DE COMMANDE

→ panier dynamique
→ formulaire de paiement complet
→ confirmation et historique

STRUCTURE DU

PROJET

Authentification

→ login.php
→ register.php
→ logout.php

PANIER

→ cart.php
→ add_to_cart.php
→ update_cart.php
→ remove_from_cart.php

COMMANDES

→ payment.php
→ process_payment.php
→ order_confirmation.php
→ orders.php

COMPTE

→ setting.php
→ orders.php

FRONTEND

→ styles.css
→ script.js

CONFIG

→ config.php

PANIER

DYNAMIQUE

Notre panier utilise les sessions PHP pour stocker temporairement les articles.

C'est un tableau associatif qui lie chaque ID de produit à sa quantité. Cette méthode est rapide car elle évite les requêtes SQL répétées, et les données restent en mémoire pendant toute la session.

Le transfert vers la base de données ne se fait qu'au moment de la validation de la commande, ce qui optimise les performances du site.

💾 Persistant pendant la session
⚡ Modifications instantanées

```php
require_once 'config.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $productId = $_POST['product_id'] ?? 0;

    if ($productId > 0) {
        if (!isset($_SESSION['cart'])) {
            $_SESSION['cart'] = [];
        }

        if (isset($_SESSION['cart'][$productId])) {
            $_SESSION['cart'][$productId]++;
        } else {
            $_SESSION['cart'][$productId] = 1;
        }
    }
}

header('Location: index.php');
exit();
?>
```

✅ Sessions PHP = stockage côté serveur
✅ Tableau associatif = structure ID → quantité
✅ Persistant pendant la session
✅ Performances optimisées (pas de requêtes SQL constantes)
✅ Transfert BDD uniquement à la validation

FONCTIONNALITÉS
✅ Ajouter un produit
✅ Modifier la quantité (+/-)
✅ Supprimer un article
✅ Calcul du total en temps réel

TUNEL D’ACHAT COMPLET

PROCESSUS DE COMMANDE

PANIER

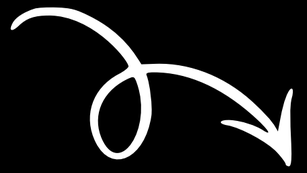
→ Validation des articles
→ Calcul du total

PAIEMENT (payment.php)

→ Informations personnelles
→ Adresse de livraison
→ Coordonnées bancaires

TRAITEMENT (process_payment.php)

→ Transaction SQL sécurisée
→ Mise à jour du stock
→ Enregistrement dans BDD



CONFIRMATION (order_confirmation.php)

→ Récapitulatif complet
→ Numéro de commande
→ Estimation de livraison

GESTION DU COMPTE UTILISATEURS

FONCTIONALITÉES

📝 MODIFIER LE PROFIL :

→ Nom d'utilisateur
→ Email
→ Vérification d'unicité

🔐 CHANGER LE MOT DE PASSE :

→ Vérification de l'ancien
→ Indicateur de force
→ Re-cryptage avec bcrypt

🗑️ SUPPRIMER LE COMPTE :

→ Modal de confirmation
→ Vérification par mot de passe
→ Suppression définitive

GESTION DU COMPTE UTILISATEURS

FONCTIONALITÉES

📝 MODIFIER LE PROFIL :

→ Nom d'utilisateur

→ Email

→ Vérification d'unicité

```php
// Traitement du formulaire de mise à jour du profil
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['action'])) {

    if ($_POST['action'] === 'update_profile') {
        $new_username = trim($_POST['username'] ?? '');
        $new_email = trim($_POST['email'] ?? '');

        if (empty($new_username) || empty($new_email)) {
            $error = 'Tous les champs sont obligatoires';
        } elseif (strlen($new_username) < 3) {
            $error = 'Le nom d\'utilisateur doit contenir au moins 3 caractères';
        } elseif (!filter_var($new_email, FILTER_VALIDATE_EMAIL)) {
            $error = 'Email invalide';
        } else {
            // Vérifier si le nom d'utilisateur ou l'email existe déjà (sauf pour l'utilisateur actuel)
            $stmt = $conn->prepare("SELECT id FROM users WHERE (username = ? OR email = ?) AND id != ?");
            $stmt->execute([$new_username, $new_email, $_SESSION['user_id']]);
            $existingUser = $stmt->fetch(PDO::FETCH_ASSOC);

            if ($existingUser) {
                $error = 'Ce nom d\'utilisateur ou cet email est déjà utilisé';
            } else {
                try {
                    $stmt = $conn->prepare("UPDATE users SET username = ?, email = ? WHERE id = ?");
                    $stmt->execute([$new_username, $new_email, $_SESSION['user_id']]);

                    $_SESSION['username'] = $new_username;
                    $user['username'] = $new_username;
                    $user['email'] = $new_email;

                    $success = 'Profil mis à jour avec succès !';
                } catch (PDOException $e) {
                    $error = 'Erreur lors de la mise à jour';
                }
            }
        }
    }
```

GESTION DU COMPTE UTILISATEURS

FONCTIONALITÉES

📝 MODIFIER LE PROFIL :

→ Nom d'utilisateur
→ Email
→ Vérification d'unicité

🔐 CHANGER LE MOT DE PASSE :

→ Vérification de l'ancien
→ Indicateur de force
→ Re-cryptage avec bcrypt

🗑️ SUPPRIMER LE COMPTE :

→ Modal de confirmation
→ Vérification par mot de passe
→ Suppression définitive

GESTION DU COMPTE UTILISATEURS

FONCTIONALITÉES

🔐 CHANGER LE MOT DE PASSE :

→ Vérification de l'ancien

→ Indicateur de force

→ Re-cryptage avec bcrypt

```php
// Changement de mot de passe
elseif ($_POST['action'] === 'change_password') {
    $current_password = $_POST['current_password'] ?? '';
    $new_password = $_POST['new_password'] ?? '';
    $confirm_password = $_POST['confirm_password'] ?? '';

    if (empty($current_password) || empty($new_password) || empty($confirm_password)) {
        $error = 'Tous les champs sont obligatoires';
    } elseif (!password_verify($current_password, $user['password'])) {
        $error = 'Mot de passe actuel incorrect';
    } elseif (strlen($new_password) < 6) {
        $error = 'Le nouveau mot de passe doit contenir au moins 6 caractères';
    } elseif ($new_password !== $confirm_password) {
        $error = 'Les nouveaux mots de passe ne correspondent pas';
    } else {
        try {
            $hashedPassword = password_hash($new_password, PASSWORD_DEFAULT);
            $stmt = $conn->prepare("UPDATE users SET password = ? WHERE id = ?");
            $stmt->execute([$hashedPassword, $_SESSION['user_id']]);

            $success = 'Mot de passe modifié avec succès !';
        } catch (PDOException $e) {
            $error = 'Erreur lors du changement de mot de passe';
        }
    }
}
```

GESTION DU COMPTE UTILISATEURS

FONCTIONALITÉES

📝 MODIFIER LE PROFIL :

→ Nom d'utilisateur
→ Email
→ Vérification d'unicité

🔐 CHANGER LE MOT DE PASSE :

→ Vérification de l'ancien
→ Indicateur de force
→ Re-cryptage avec bcrypt

🗑️ SUPPRIMER LE COMPTE :

→ Modal de confirmation
→ Vérification par mot de passe
→ Suppression définitive

GESTION DU COMPTE UTILISATEURS

FONCTIONALITÉES

🗑️ SUPPRIMER LE COMPTE :

→ Modal de confirmation

→ Vérification par mot de passe

→ Suppression définitive

```php

    // Suppression du compte
    elseif ($_POST['action'] === 'delete_account') {
        $password_confirm = $_POST['password_confirm'] ?? '';

        if (!password_verify($password_confirm, $user['password'])) {
            $error = 'Mot de passe incorrect';
        } else {
            try {
                $stmt = $conn->prepare("DELETE FROM users WHERE id = ?");
                $stmt->execute([$_SESSION['user_id']]);

                session_destroy();
                header('Location: index.php?account_deleted=1');
                exit();
            } catch (PDOException $e) {
                $error = 'Erreur lors de la suppression du compte';
            }
        }
    }
}
```

GROUPE FRAME

WWW.REALLYGREATSITE.COM

PRESENTATION DU SITE

EN VIDEO

22/08/29

VIDEO EN PIECE JOINTE DU MAIL

CONSLUSION

OBJECTIF

OBJECTIFS ATTEINTS ✅

→ Site e-commerce fonctionnel
→ Base de données normalisée
→ Sécurité renforcée
→ Expérience utilisateur fluide
→ visualisation des stocks

A Corriger :

→ suppression du compte

A Ajouter :

→ Compte admin pour gerer les produit les stock etc...

CONCLUSION

SATISTIQUES

📂 ARCHITECTURE

→ 16 fichiers PHP
→ 5 tables SQL
→ 8 relations de clés étrangères

⚡ FONCTIONNALITÉS

→ 8 produits en catalogue
→ Panier temps réel
→ Système de commande complet
→ gestion du compte utilisateur
→ bisualisation de toutes les commandes