

Projet E6 – Site e-commerce Persona WORLD – en cours

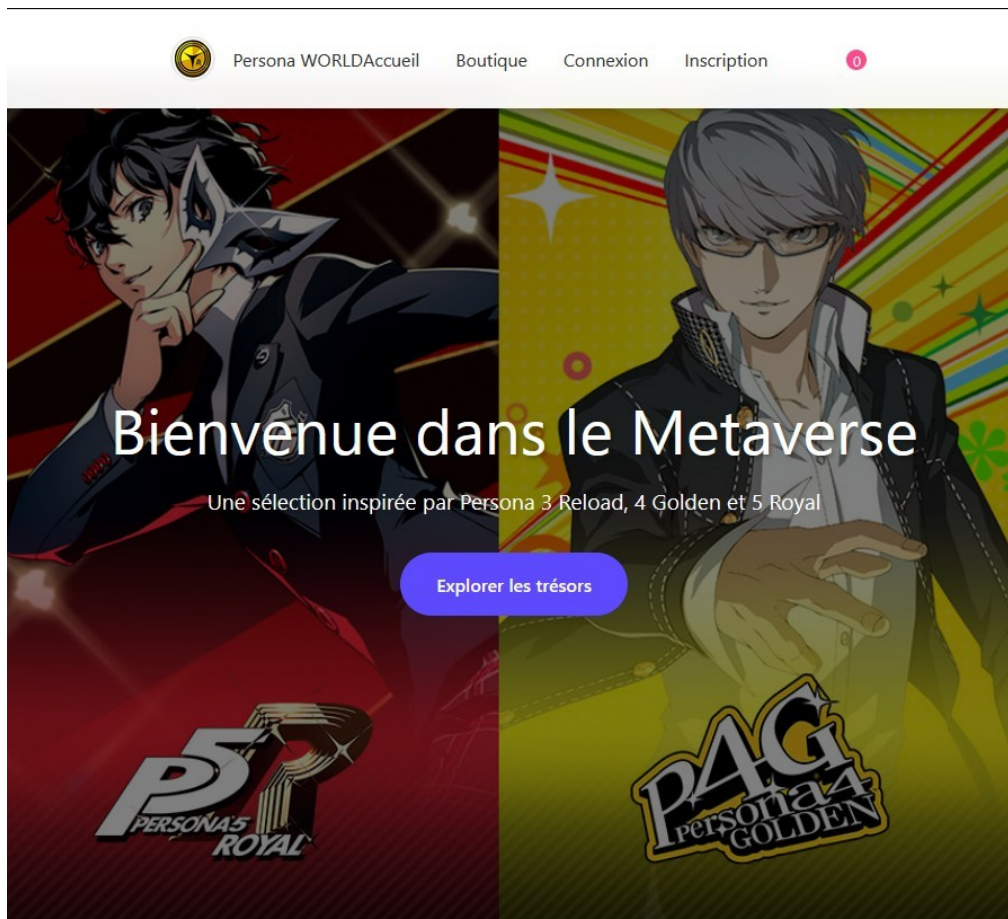
Présentation du projet

Ce projet est un site e-commerce fictif inspiré de l'univers des jeux Persona. Il permet aux utilisateurs de consulter des produits, de s'inscrire, de se connecter et d'accéder à leur espace personnel. Le site utilise PHP avec Twig, une base de données MySQL, et fonctionne localement avec WAMP.

Fonctionnalités développées

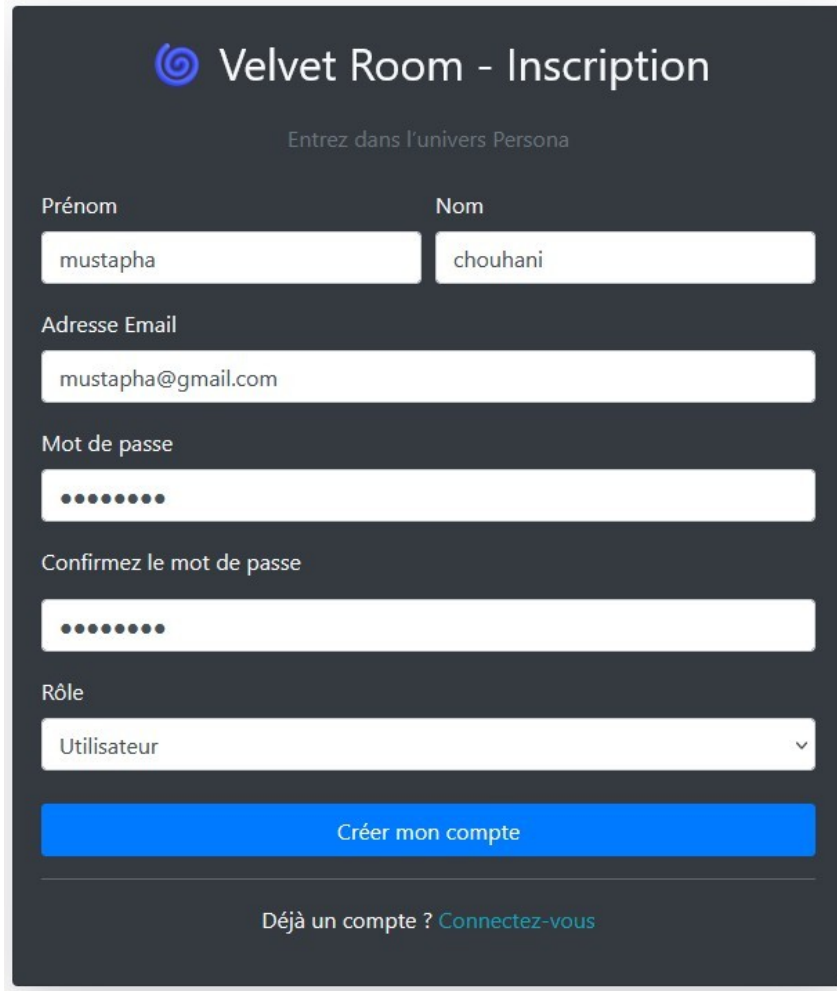
1. Accueil

La page d'accueil présente les catégories de produits (figurines, vêtements, accessoires) ainsi qu'une sélection d'articles en vedette. L'utilisateur peut naviguer dans le site via le menu en haut de page.



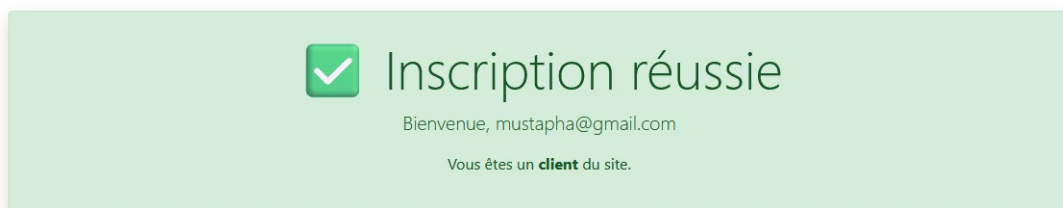
2. Inscription

Le formulaire d'inscription permet aux nouveaux utilisateurs de créer un compte avec leur nom, prénom, adresse email, mot de passe et rôle. Le mot de passe est sécurisé avec un hash (bcrypt).



The image shows a registration form for 'Velvet Room'. The form is titled 'Velvet Room - Inscription' with a subtitle 'Entrez dans l'univers Persona'. It contains several input fields: 'Prénom' (first name) with the value 'mustapha', 'Nom' (last name) with the value 'chouhani', 'Adresse Email' with the value 'mustapha@gmail.com', 'Mot de passe' (password) and 'Confirmez le mot de passe' (confirm password) both masked with dots, and a 'Rôle' (role) dropdown menu currently set to 'Utilisateur'. A blue button labeled 'Créer mon compte' is at the bottom of the form. Below the button, there is a link 'Déjà un compte ? Connectez-vous'.

← → ↺ http://localhost/shop-shop - Copie/public/index.php?page=inscrire ☆ 🔒 ⬇️ ⓘ 📄 ☰



Extrait du code d'inscription (PHP) :

```
<?php
0 references | Windsurf: Refactor | Explain | Generate Function Comment | X
function inscrireControlleur($twig) : void {
    $form = array();

    if (isset($_POST['btnInscrire'])) {
        $inputEmail = filter_input(type: INPUT_POST, var_name: 'inputEmail', filter: FILTER_VALIDATE_EMAIL);
        $inputPassword = $_POST['inputPassword'];
        $inputPassword2 = $_POST['inputPassword2'];
        $nom = htmlspecialchars(string: $_POST['nom']);
        $prenom = htmlspecialchars(string: $_POST['prenom']);
        $role = $_POST['role'];

        if (!$inputEmail) {
            $form['valide'] = false;
            $form['message'] = 'Email invalide';
        }

        else if ($inputPassword !== $inputPassword2) {
            $form['valide'] = false;
            $form['message'] = 'Les mots de passe sont différents';
        }

        else if (strlen(string: $inputPassword) < 8) {
            $form['valide'] = false;
            $form['message'] = 'Le mot de passe doit contenir au moins 8 caractères';
        }

    }

    else {
        try {
            $pdo = new PDO(dsn: "mysql:host=localhost;dbname=PeW;charset=utf8", username: "root", password: "");
            $pdo->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);


            $checkSql = "SELECT COUNT(*) FROM utilisateur WHERE email = ?";
            $checkStmt = $pdo->prepare(query: $checkSql);
            $checkStmt->execute(params: [$inputEmail]);

            if ($checkStmt->fetchColumn() > 0) {
                $form['valide'] = false;
                $form['message'] = 'Cet email est déjà utilisé';
            } else {
                $sql = "INSERT INTO utilisateur (prenom, nom, email, password, idRole) VALUES (?, ?, ?, ?, ?)";
                $stmt = $pdo->prepare(query: $sql);
                $stmt->execute(params: [
                    $prenom,
                    $nom,
                    $inputEmail,
                    password_hash(password: $inputPassword, algo: PASSWORD_DEFAULT),
                    $role
                ]);
                $form['valide'] = true;
                $form['email'] = $inputEmail;
                $form['role'] = $role;
            }
        } catch (PDOException $e) {
            $form['valide'] = false;
            $form['message'] = "Erreur lors de l'inscription : " . $e->getMessage();
        }
    }

    echo $twig->render('inscrire.html.twig', array('form' => $form));
}
```

3. Connexion

Le formulaire de connexion permet à l'utilisateur de se connecter à son compte grâce à son adresse email et son mot de passe. Une vérification est faite via ``password_verify()`` sur le mot de passe haché.



Connexion à la Velvet Room


Identifiez-vous pour accéder à votre collection Persona

Adresse Email

Mot de passe

[Se connecter](#)

Pas encore de compte ? [Inscrivez-vous](#)



Connexion réussie

Bienvenue, mustapha@gmail.com

Vous êtes un **client** du site.

Extrait du code de connexion (PHP) :

```
<?php
0 references | Windsurf: Refactor | Explain | Generate Function Comment | X
function connexionControleur($twig) : void {

    $form = [];

    if (isset($_POST['btnConnexion'])) {
        $email = $_POST['inputEmail'];
        $password = $_POST['inputPassword'];

        // Connexion à la BDD
        try {
            $pdo = new PDO(dsn: "mysql:host=localhost;dbname=PeW;charset=utf8", username: "root", password: "");
            $sql = "SELECT * FROM utilisateur WHERE email = ?";
            $stmt = $pdo->prepare(query: $sql);
            $stmt->execute(params: [$email]);
            $user = $stmt->fetch();

            if ($user && password_verify(password: $password, hash: $user['password'])) {
                $form['valide'] = true;
                $form['email'] = $user['email'];
                $form['role'] = $user['idRole'];
            } else {
                $form['valide'] = false;
                $form['message'] = "Identifiants incorrects.";
            }
        } catch (PDOException $e) {
            $form['valide'] = false;
            $form['message'] = "Erreur serveur : " . $e->getMessage();
        }
    }

    echo $twig->render('connexion.html.twig', ['form' => $form]);
}
```

4. Base de données

Voici les utilisateurs enregistrés dans la base de données MySQL après inscription réussie.

id	email	password	nuser	nom	prenom
5	test@gmail.com	\$2b\$10\$uWx3tAIPjhw/eXJr2z3NE.Bsy1jAzhXA85OZOBnw9eF...		test	test
7	mustapha@gmail.com	\$2y\$10\$GMFflhPgGUbT0Y8Q0JI62u7cemRBhvEIZcD7sf4l8uD...		chouhani	mustapha

5. Arborescence du projet

Voici la structure du projet, montrant l'organisation des fichiers dans le dossier principal ('shop-shop') et les sous-dossiers : config, src (avec controleur, modele, vue), etc.





Index of /shop-shop

Name	Last modified	Size	Description
 Parent Directory		-	
 config/	2025-04-18 21:36	-	
 lib/	2025-04-18 21:36	-	
 maquettes/	2024-11-02 12:29	-	
 node_modules/	2025-04-18 21:36	-	
 package-lock.json	2025-01-23 11:36	56K	
 package.json	2025-01-23 11:36	393	
 public/	2025-04-18 23:32	-	
 script.js	2025-01-23 10:38	1.3K	
 src/	2025-04-18 21:36	-	

Index of /shop-shop/config

Name	Last modified	Size	Description
 Parent Directory		-	
 connexion.php	2025-04-18 22:00	351	
 parametres.php	2025-04-18 22:01	120	
 routes.php	2025-01-23 08:24	738	

Index of /shop-shop/src

Name	Last modified	Size	Description
 Parent Directory		-	
 controleur/	2025-04-18 21:36	-	
 modele/	2025-04-18 21:36	-	
 vue/	2025-04-18 21:36	-	

Index of /shop-shop/src/contrôleur

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 _contrôleurs.php	2025-04-18 23:36	184	
 accueilContrôleur.php	2025-01-23 08:24	224	
 connexionContrôleur.php	2025-04-19 00:04	1.1K	
 deconnexionContrôleur.php	2025-01-30 09:39	125	
 inscrireContrôleur.php	2025-04-19 00:27	2.5K	
 utilisateurContrôleur.php	2025-01-30 11:30	263	

Index of /shop-shop/src/vue

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 accueil.html.twig	2025-04-19 00:13	6.2K	
 base.html.twig	2025-04-18 23:00	1.7K	
 connexion.html.twig	2025-04-19 00:08	3.0K	
 inscrire.html.twig	2025-04-18 23:44	4.0K	
 inscrire.html.twig	2024-11-21 12:17	1.6K	
 maintenance.html.twig	2025-01-23 08:24	201	
 utilisateur.html.twig	2025-01-30 13:22	380	

Fichier principal index.php utilisé pour charger Twig et diriger les pages :

```
<?php
session_start();

require_once '../src/contrôleur/_contrôleurs.php';
require_once '../lib/vendor/autoload.php';
require_once '../config/parametres.php';
require_once '../config/connexion.php';
require_once '../config/routes.php';
require_once '../src/modele/_classes.php';

$loader = new \Twig\Loader\FilesystemLoader(paths: '../src/vue/');
$twig = $twig = new \Twig\Environment(loader: $loader, options: []);
// $twig->addGlobal('session', $_SESSION);
$db = connect(config: $config);
$contenu = getPage(db: $db);
$contenu($twig, $db);
```