

Livrable semaine 50

User Stories prioritaires réalisées



Table des matières

Introduction.....	3
1. Créer un compte.....	4
1. Script.....	4
2. Trace d'exécution.....	7
2. Se connecter à un compte.....	8
1. Script.....	8
2. Trace d'exécution.....	11
3. Se déconnecter d'un compte.....	13
1. Script.....	13
2. Trace d'exécution.....	13
4. Chercher un produit par mot-clé.....	14
1. Script.....	14
2. Trace d'exécution.....	14
5. Chercher un produit selon des critères de tris.....	15
1. Script.....	15
2. Trace d'exécution.....	24
6. Consulter un produit.....	26
1. Script.....	26
2. Trace d'exécution.....	31
7. Ajout d'un produit au panier.....	33
1. Script.....	33
2. Trace d'exécution.....	36
8. Consulter le panier.....	37
1. Script.....	37
2. Trace d'exécution.....	39
Conclusion.....	41

Introduction

Ce livrable regroupe l'ensemble des scripts et des traces d'exécution réalisés pour les user stories prioritaires du projet. Il a pour objectif de présenter clairement le fonctionnement des principales fonctionnalités attendues et de montrer, à travers des preuves concrètes, que celles-ci répondent correctement aux besoins définis. Chaque section associe un script et sa trace d'exécution, ce qui permet d'illustrer à la fois la démarche suivie et le résultat obtenu.

Le document suit l'ordre logique d'utilisation de la plateforme : il commence par la création d'un compte, puis enchaîne avec la connexion et la déconnexion de l'utilisateur. Il présente ensuite les différentes méthodes de recherche de produits, qu'il s'agisse d'une recherche par mot-clé, par catégorie, sous-catégorie, regroupements ou encore par prix. La consultation d'un produit est détaillée, incluant les informations techniques, les photos, les avis clients et les produits apparentés. Enfin, l'ajout au panier et la gestion du panier (modification des quantités, suppression d'articles) sont présentés.

1. Créer un compte

[\[US 01\]: En tant que visiteur, je veux créer un compte](#)

1. Script

```
<?php
session_start();
require_once 'connec.inc.php';

// Redirection si accès direct sans POST
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    header('Location: connexion.php');
    exit;
}

// Récupération des champs du formulaire
$prenom      = isset($_POST['prenom']) ? trim($_POST['prenom']) : '';
$nom        = isset($_POST['nom']) ? trim($_POST['nom']) : '';
$email       = isset($_POST['email']) ? trim($_POST['email']) : '';
$motDePasse = isset($_POST['motDePasse']) ? $_POST['motDePasse'] : '';
$confirmation = isset($_POST['confirmation']) ? $_POST['confirmation']
: '';
$adresse      = isset($_POST['adresse']) ? trim($_POST['adresse']) : '';
$codePostal   = isset($_POST['codePostal']) ? trim($_POST['codePostal']) :
: '';
$ville        = isset($_POST['ville']) ? trim($_POST['ville']) : '';
$telephone     = isset($_POST['telephone']) ? trim($_POST['telephone']) :
: '';

$erreurs = [];

// Vérifications du remplissage des champs "obligatoires"
if ($prenom === '' || $nom === '' || $email === '' || $motDePasse ===
'' || $confirmation === '') ||
($adresse === '' || $codePostal === '' || $ville === '')) {
    $erreurs[] = "Tous les champs obligatoires doivent être remplis.";
}

// Validation de l'email (au format requis)
if (!preg_match('/^([a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$/',
$email)) {
```

```

$erreurs[] = "L'adresse e-mail n'est pas valide.";
}

if ($motDePasse !== $confirmation) {
    $erreurs[] = "Les mots de passe ne correspondent pas.";
}

// Validation du mot de passe (au format requis)
if
(!preg_match('/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^a-zA-Z\d]).{8,}$/', $motDePasse)) {
    $erreurs[] = "Le mot de passe doit contenir au moins 8 caractères, incluant une lettre minuscule, une majuscule, un chiffre et un caractère spécial.";
}

// Validation du numéro de téléphone (au format requis)
if ($telephone !== '' && !preg_match('/^([0-9]{10})$/', $telephone)) {
    $erreurs[] = "Le numéro de téléphone doit contenir exactement 10 chiffres.";
}

// Vérifier si l'e-mail existe déjà dans la base de données
if (empty($erreurs)) {
    try {
        $stmt = $connection->prepare("SELECT COUNT(*) FROM Client WHERE email = :email");
        $stmt->bindParam(':email', $email, PDO::PARAM_STR);
        $stmt->execute();
        $existe = $stmt->fetchColumn();

        if ($existe > 0) {
            $erreurs[] = "Un compte existe déjà avec cette adresse e-mail.";
        }
    } catch (PDOException $e) {
        $erreurs[] = "Erreur lors de la vérification de l'e-mail.";
    }
}

// S'il y a des erreurs, renvoie vers la page de connexion/inscription avec un message d'erreur
if (!empty($erreurs)) {

```

```

$msg = implode(' ', $erreurs);
header("Location: connexion.php?msgErreur=" . urlencode($msg) .
"&mode=register");
exit;
}

// Insertion du nouveau client
try {
    $hash = password_hash($motDePasse, PASSWORD_DEFAULT);

    $stmt = $connection->prepare("
        INSERT INTO Client (nom, prenom, email, motDePasse, adresse,
codePostal, ville, telephone)
        VALUES (:nom, :prenom, :email, :motDePasse, :adresse,
:codePostal, :ville, :telephone)
    ");

    $stmt->bindParam(':nom', $nom, PDO::PARAM_STR);
    $stmt->bindParam(':prenom', $prenom, PDO::PARAM_STR);
    $stmt->bindParam(':email', $email, PDO::PARAM_STR);
    $stmt->bindParam(':motDePasse', $hash, PDO::PARAM_STR);
    $stmt->bindParam(':adresse', $adresse, PDO::PARAM_STR);
    $stmt->bindParam(':codePostal', $codePostal, PDO::PARAM_STR);
    $stmt->bindParam(':ville', $ville, PDO::PARAM_STR);
    // téléphone peut être NULL
    if ($telephone === '') {
        $stmt->bindValue(':telephone', null, PDO::PARAM_NULL);
    } else {
        $stmt->bindParam(':telephone', $telephone, PDO::PARAM_STR);
    }

    $stmt->execute();

    // On peut soit connecter directement l'utilisateur, soit le
    renvoyer à la page de connexion
    header("Location: connexion.php?msgSucces=" . urlencode("Compte
créé avec succès, vous pouvez vous connecter."));
    exit;
} catch (PDOException $e) {
    header("Location: connexion.php?msgErreur=" . urlencode("Erreur
lors de la création du compte."));
    exit;
}

```

2. Trace d'exécution

Affichage du menu de création :

Succès de la création du compte :

Connexion des Aventuriers

Compte créé avec succès, vous pouvez vous connecter.

Echec de la création du compte si l'email est déjà utilisé:

Création de Compte

Un compte existe déjà avec cette adresse e-mail.

Echec de la création du compte si le mot de passe ne respecte pas le format attendu :

Création de Compte

Le mot de passe doit contenir au moins 8 caractères, incluant une lettre minuscule, une majuscule, un chiffre et un caractère spécial.

2. Se connecter à un compte

[US 02]: En tant qu'utilisateur, je veux me connecter avec login et mot de passe

1. Script

Script du traitement de la connexion

```
<?php
session_start();

require_once 'connec.inc.php';

// --- Fonctions de vérification ---
function verifierEmail($email) {
    global $connection;
    $stmt = $connection->prepare("SELECT * FROM Client WHERE email = :email");
    $stmt->bindParam(':email', $email);
    $stmt->execute();
    return $stmt->fetch(PDO::FETCH_ASSOC) ?: false;
}

function verifierMotDePasse($email, $motDePasse) {
    global $connection;
    $stmt = $connection->prepare("SELECT motDePasse FROM Client WHERE email = :email");
    $stmt->bindParam(':email', $email);
    $stmt->execute();
    $hash = $stmt->fetchColumn();
    return $hash && password_verify($motDePasse, $hash);
}

function getInfosClient($email, $db) {
    $sql = "SELECT idClient, nom, prenom FROM Client WHERE email = :email LIMIT 1";
    $stmt = $db->prepare($sql);
    $stmt->execute([':email' => $email]);
    return $stmt->fetch(PDO::FETCH_ASSOC);
}

// --- Traitement ---
```

```
if (!isset($_POST['login'], $_POST['login-password'])) {
    header("Location: connexion.php?msgErreur=" . urlencode("Formulaire
incomplet"));
    exit;
}

$email = $_POST['login'];
$pwd   = $_POST['login-password'];

if (verifierEmail($email) && verifierMotDePasse($email, $pwd)) {

    $_SESSION["SLeRoiMerlin"] = "OK";
    $infosClient = getInfosClient($email, $connection);

    if ($infosClient) {
        $_SESSION["idClient"] = $infosClient['idClient'];
        $_SESSION["nomClient"] = $infosClient['nom'];
        $_SESSION["prenomClient"] = $infosClient['prenom'];
    }

    // GESTION DU TEMPS
    if (!empty($_POST["remember-me"])) {
        $duree_cookie = 7200;

        // 1. On crée le cookie physique
        setcookie("CLeRoiMerlin", $email, time() + $duree_cookie, "/");

        // 2. On note l'heure précise de la mort dans la session
        $_SESSION['heure_expiration_stricte'] = time() + $duree_cookie;
    }

    } else {
        // Si pas de "Se souvenir de moi", on supprime les traces
        unset($_SESSION['heure_expiration_stricte']);
        setcookie("CLeRoiMerlin", "", time() - 3600, "/");
    }

    header("Location: index.php");
    exit;
} else {
    header("Location: connexion.php?msgErreur=" .
urlencode("Identifiants incorrects"));
    exit;
}
?>
```

Script du traitement de la re connexion automatique via le cookie

```
// --- 1. SÉCURITÉ ABSOLUE : VÉRIFICATION DU TEMPS ---
// Si une date d'expiration est fixée ET qu'on l'a dépassée -> DEHORS
if (isset($_SESSION['heure_expiration_stricte']) && time() >
$_SESSION['heure_expiration_stricte']) {

    // Destruction totale
    $_SESSION = [];
    if (ini_get("session.use_cookies")) {
        $params = session_get_cookie_params();
        setcookie(session_name(), '', time() - 42000, $params["path"],
$params["domain"], $params["secure"], $params["httponly"]);
    }
    setcookie("CLeRoiMerlin", '', time() - 3600, "/");
    session_destroy();

    header("Location: connexion.php?msgErreur=" . urlencode("Votre
session a expiré (Temps écoulé)."));
    exit;
}

// --- 2. RECONNEXION AUTO (Si le cookie est encore valide et qu'on
n'est pas connecté) ---
if (!isset($_SESSION["idClient"]) && isset($_COOKIE["CLeRoiMerlin"])) {

    $emailCookie = $_COOKIE["CLeRoiMerlin"];

    if (isset($connection)) {
        $stmt = $connection->prepare("SELECT idClient, nom, prenom FROM
Client WHERE email = :email");
        $stmt->execute([':email' => $emailCookie]);
        $clientTrouve = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($clientTrouve) {
            $_SESSION["SLeRoiMerlin"] = "OK";
            $_SESSION["idClient"] = $clientTrouve['idClient'];
            $_SESSION["prenomClient"] = $clientTrouve['prenom'];
            $_SESSION["nomClient"] = $clientTrouve['nom'];
            $duree_prolongation = 15; // Mettre la même valeur que dans
traitement.php
            setcookie("CLeRoiMerlin", $emailCookie, time() +
$duree_prolongation, "/");
        }
    }
}
```

```

        $_SESSION['heure_expiration_stricte'] = time() +
$duree_prolongation;
    }
}
}

// --- 3. VERIFICATION DE LA DISPARITION DU COOKIE ---
// Si on devait avoir un cookie (heure_expiration_stricte existe) MAIS
que le navigateur ne l'envoie plus
if (isset($_SESSION["idClient"]) &&
isset($_SESSION['heure_expiration_stricte']) &&
empty($_COOKIE["CLeRoiMerlin"])) {
    // Cas rare où le navigateur a supprimé le cookie avant PHP
    $_SESSION = [];
    session_destroy();
    header("Location: connexion.php?msgErreur=" . urlencode("Session
expirée."));
    exit;
}

```

2. Trace d'exécution

Affichage de la page de connexion au compte :

The screenshot shows a login form titled "Connexion des Aventuriers". It has two tabs at the top: "Se Connecter" (highlighted in green) and "Créer un Compte". Below the tabs are two input fields: "E-mail" and "Mot de Passe", each with a corresponding empty text input box. There is also a checkbox labeled "Se souvenir de moi". At the bottom is a large green button labeled "Se Connecter".

Erreur affichée en cas de mot de passe incorrect :

Connexion des Aventuriers

Identifiants incorrects

Succès de la connexion au compte : le nom d'utilisateur est affiché en haut à droite



Le Roi Merlin
Artisanat Magique

Accueil

Catalogue

Mon Compte (Antonin)



3. Se déconnecter d'un compte

[US 03]: En tant qu'utilisateur, je veux me déconnecter

1. Script

```
<?php  
// 1. On démarre la session pour pouvoir la manipuler  
session_start();  
$_SESSION = array();  
  
if (ini_get("session.use_cookies")) {  
    $params = session_get_cookie_params();  
    setcookie(session_name(), '', time() - 42000,  
              $params["path"], $params["domain"],  
              $params["secure"], $params["httponly"]  
    );  
}  
setcookie("CLeRoiMerlin", '', time() - 7200, "/");  
session_destroy();  
header("Location: index.php");  
exit;  
?>
```

2. Trace d'exécution

Depuis le menu de gestion du compte on clique sur "Déconnexion"

Tableau de Bord de l'Aventurier



The screenshot shows the 'Tableau de Bord de l'Aventurier' interface. On the left, a sidebar lists navigation options: 'Mon Tableau de Bord' (highlighted in green), 'Mes Commandes', 'Mes Informations', and 'Déconnexion'. A red arrow points to the 'Déconnexion' link. The main content area displays a welcome message: 'Bienvenue, Antonin Le donné !' followed by a brief description: 'C'est ici que vous gérez vos sorts, vos achats et vos informations personnelles.' Below this, a section titled 'Dernières Activités' shows three metrics: '3 Commandes Passées', '0 Points de Magie (Fidélité)', and 'Commande #108 Payée Du 09/12/2025'. A yellow button at the bottom right of this section says 'Voir toutes mes commandes'.

Le nom d'utilisateur est remplacé par "Se Connecter". L'utilisateur n'est plus connecté.



4. Chercher un produit par mot-clé

[US 04]: En tant que visiteur, je veux chercher un produit par mot-clé

1. Script

```
<?php  
$search = trim($_GET['search'] ?? ''); //Récupérer le param de la  
recherche si il existe sinon chaine vide  
  
if ($search === '') { //Si aucune recherche on affiche le catalogue  
normal  
    header('Location: catalogue.php');  
    exit;  
}  
  
// Redirection vers le catalogue avec le paramètre search  
header('Location: catalogue.php?search=' . urlencode($search));  
exit;
```

2. Trace d'exécution

On recherche un élément

A search bar with a rounded rectangular border and a light green gradient. Inside the bar, the text 'Potion de' is written in a dark blue font. To the right of the text is a magnifying glass icon.

Les produits correspondants à la recherche sont affichés dans le catalogue

The screenshot shows the Le Roi Merlin website's catalogue page. At the top, there is a navigation bar with links for Accueil, Catalogue, Mon Compte (tux), and a shopping cart icon showing 1 item. Below the navigation is a search bar with the placeholder "Potion de". The main content area is titled "Catalogue" and features a grid of four product cards. Each card includes a small image, the product name, its rarity level (e.g., "Commun", "Peu commun"), a brief description, the price (e.g., 19.99€, 7.99€, 9.99€), and two buttons: "Voir les Détails" and "Ajouter au panier".

Produit	Rareté	Description	Prix
Potion de mana majeure	Commun	Potion de mana majeure Potion de mana	19.99€
Potion de mana mineure	Commun	Potion de mana mineure Potion de mana	7.99€
Potion de Résistance à la foudre	Peu commun	Potion de Résistance à la foudre Potion de résistance	9.99€
Potion de Résistance à la glace	Peu commun	Potion de Résistance à la glace Potion de résistance	9.99€

5. Chercher un produit selon des critères de tris

User stories concernées :

[US 05]: En tant que visiteur, je veux chercher un produit par catégorie / sous-catégorie

[US 06]: En tant que visiteur, je veux rechercher par regroupement

[US 08]: En tant que visiteur, je veux trier la liste des produits par prix croissant ou décroissant

[US07] : En tant que visiteur, je veux visualiser la liste des produits correspondant à ma recherche, afin de comparer et choisir.

1. Script

```
<?php

    require_once("./connec.inc.php");
    // Si le fichier est exécuté directement (en tapant le nom du
fichier dans l'url)
    if (basename($_SERVER['SCRIPT_FILENAME']) === basename(__FILE__)) {
        header("Location: catalogue.php"); //On redirige vers le
catalogue
        exit;
    }

    //On récup tous les params de la requête
    $quality      = $_GET['quality'] ?? null; //Si isset quality alors
on le récup, sinon on met null
```

```

$regroupement = $_GET['regroupement'] ?? null;
$category      = $_GET['category'] ?? null;
$subCategory   = $_GET['sub-category'] ?? null;
$sort          = $_GET['sort'] ?? null;
$search        = $_GET['search'] ?? null;

//On charge tous les éléments nécessaires
$qualites = chargerQualites();
$regroupements = chargerRegroupements();
$categories = chargerCategories();
$produits = effectuerTriProduits( //On effectue le tri des produits
à afficher
    $quality,
    $regroupement,
    $category,
    $subCategory,
    $sort,
    $search
);

/***
 * Fonction permettant de charger les différentes qualités
disponibles
 */

function chargerQualites()
{
    $tab_qualites = [];
    try {
        global $connection;
        $statement = $connection->prepare("SELECT DISTINCT qualite
FROM DeclinaisonProduit"); //Récup toutes les qualités
        $statement->execute(); //Executer la commande

        while ($ligne = $statement->fetch((PDO::FETCH_ASSOC))) {
//Récup un tableau associatif
            $tab_qualites[] = $ligne['qualite'];
        }
    } catch (PDOException $e) {

```

```

        error_log("Erreur SQL chargerQualites : " .
$e->getMessage());
    }

    if (empty($tab_qualites)) { //Si aucune qualité n'est trouvée
        $tab_qualites[] = 'Aucune qualité trouvée';
    }

    return $tab_qualites; //On renvoie le tableau de qualités
}

/**
 * Fonction permettant de charger les différentes regroupements
disponibles
 */
function chargerRegroupements()
{
    $tab_regroupements = [];

    try {
        global $connection;
        $statement = $connection->prepare("SELECT * FROM
Regroupement"); //Récup tous les regroupements
        $statement->execute(); //Executer la commande

        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
//Récup un tableau associatif
            $tab_regroupements[$ligne['idRegroupement']] =
$ligne['libelle']; //On enregistre idRegroupement => libelle pour chaque
regroupement
        }
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerRegroupements : " .
$e->getMessage());
    }

    if (empty($tab_regroupements)) { //Si aucun regroupement n'est
trouvé
        $tab_regroupements[] = 'Aucun regroupement trouvé';
    }
}

```

```

        return $tab_regroupements; //On renvoie le tableau de
regroupements

    }

/***
 * Fonction permettant de charger les différentes catégories
disponibles
 */

function chargerCategories()
{
    $tab_categories = [];

    try {
        global $connection;
        $statement = $connection->prepare("SELECT * FROM Categorie
WHERE Categorie.idCategorieParent IS NULL");
        $statement->execute(); //Executer la commande
        while ($ligne = $statement->fetch((PDO::FETCH_ASSOC))) {
//Récup un tableau associatif
            $tab_categories[$ligne['idCategorie']] =
$ligne['libelle']; //On enregistre idCategorie -> libelle pour chaque
catégorie
        }
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerCategories : " .
$e->getMessage());
    }

    if (empty($tab_categories)) { //Si aucune catégorie n'est
trouvée
        $tab_categories[] = 'Aucune catégorie trouvée';
    }

    return $tab_categories; //On renvoie le tableau de catégories
}

/***
 * Fonction permettant de charger les différentes catégories
enfants dispos pour une catégorie parent

```

```

        */

function chargerCategoriesEnfants($idCategorieParent)
{
    $tab_categories = [];
    try {
        global $connection;
        $statement = $connection->prepare("CALL
getCategEnfants(:idCategorieParent)");
        $statement->bindParam(':idCategorieParent',
$idCategorieParent, PDO::PARAM_INT);
        $statement->execute();

        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
            $tab_categories[$ligne['idCategorie']] =
$ligne['libelle'];
        }

        // Important : libérer le curseur si la procédure retourne
plusieurs résultats
        $statement->closeCursor();
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerCategoriesEnfants : " .
$e->getMessage());
    }

    if (empty($tab_categories)) { //Si aucune catégorie n'est
trouvée
        $tab_categories[] = 'Aucune catégorie enfant trouvée';
    }

    return $tab_categories;
}

/**
 * Permet de faire une requête en bd pour charger les produits
selon les filtres appliqués
 * Retourne un statement à parcourir
 */
function chargerProduits($quality = null, $regroupement = null,
$category = null, $subCategory = null, $sort = null, $search = null)
{
    global $connection;

```

```

    $sql = "SELECT DISTINCT d.*, c.libelle as categorie_libelle,
p.libelle as produit_libelle
    FROM DeclinaisonProduit d
    LEFT JOIN Regrouper re ON re.idProduit = d.idProduit
    LEFT JOIN Regroupement r ON re.idRegroupement =
r.idRegroupement
    LEFT JOIN Appartenir a ON a.idProduit = d.idProduit
    LEFT JOIN Produit p ON p.idProduit = d.idProduit
    LEFT JOIN Categorie c ON c.idCategorie = a.idCategorie
    WHERE 1=1"; //technique 1=1 : permet de placer les AND ...
plus facilement à la suite pas besoin de savoir qui est le premier

$params = [];

// Filtre qualité
if ($quality && $quality != 'all') {
    $sql .= " AND d.qualite = :quality";
    $params[':quality'] = $quality;
}

// Filtre regroupement
if ($regroupement && $regroupement != 'all') {
    $sql .= " AND re.idRegroupement = :regroupement";
    $params[':regroupement'] = $regroupement;
}

if ($search) {
    $mots = explode(' ', $search); // Sépare la recherche par
mots
    // Exemple : si $search = "potion mana"
    // $mots = ['potion', 'mana']

    foreach ($mots as $i => $mot) { //On va comparé le mot de
la recherche à différents éléments
        $sql .= " AND (
d.libelleDeclinaison LIKE :mot$i
OR p.libelle LIKE :mot$i
OR p.description LIKE :mot$i
OR c.libelle LIKE :mot$i
OR r.libelle LIKE :mot$i
)";
    }
}

```

```

        $params[":$mot$i"] = "%$mot%";
        // Exemple après la première itération ($i=0,
$mot='potion'):
        // $params = [':mot0' => '%potion%']
        // Exemple après la deuxième itération ($i=1,
$mot='mana'):
        // $params = [':mot0' => '%potion%', ':mot1' =>
'%mana%']
    }

    // Filtre catégorie / sous-catégorie
    if ($category && $category != 'all') { //Si la catégorie est
renseignée ('' renvoie false / si la catégorie n'est pas all)
        //on prend soit l'id de la sous catégorie (si sélectionnée)
soit l'id de la catégorie parent
        $idFiltre = ($subCategory && $subCategory != 'all') ?
$subCategory : $category;
        $produits_categ = getProduitsByCategorie($idFiltre); //On
récup les ids de tous les produits de cette catégorie
        if (!empty($produits_categ)) { //Si il y a des produits
récupérés
            $placeholders = [];
            foreach ($produits_categ as $i => $idProduit) {
                $placeholder = ":prod$i"; //On crée un param nommé
pour chaque produit
                $placeholders[] = $placeholder; //Ex :
$placeholders = [":prod0", ":prod1", ":prod2"]
                //Sauvegarde pour le passage de paramètres nommés
                $params[$placeholder] = $idProduit;
                // On sauvegarde chaque paramètre nommé pour lier
l'ID du produit à la requête PDO
                // Exemple : $params = [":prod0" => 101, ":prod1"
=> 102]
            }
            //On recherche les idDeclinaisons contenues dans
$placeholders = [":prod0", ":prod1", ":prod2"]
            $sql .= " AND d.idDeclinaison IN (" . implode(',', [
$placeholders) . ")";
        }
    }

    // Tri

```

```

        if ($sort) {
            if ($sort === 'price-asc') $sql .= " ORDER BY prix ASC";
            if ($sort === 'price-desc') $sql .= " ORDER BY prix DESC";
        }
        $stmt = null; //Initialise stmt
        try {
            $stmt = $connection->prepare($sql);
            $stmt->execute($params);
        } catch (PDOException $e) {
            error_log("Erreur SQL chargerProduits : " .
$e->getMessage());
        }

        return $stmt;
    }

    /**
     * Fonction permettant d'effectuer le tri des produits
     * Retourne un tableau de produits triés
     */
    function effectuerTriProduits(
        $quality,
        $regroupement,
        $category,
        $subCategory,
        $sort,
        $search
    ) {

        $statement = chargerProduits(
            $quality,
            $regroupement,
            $category,
            $subCategory,
            $sort,
            $search
        );
        $produits = [];
        if($statement){ //Si le statement renvoyé n'est pas null
            while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
                $produits[] = $ligne; //On ajoute le produit
            }
        }
    }
}

```

```
    }

    return $produits; //Si tab produits vide retournés alors "Aucun
produit trouvé"
}

/**
 * Récupère tous les id produit d'une catégorie (ainsi que des sous
catégories)
*/
function getProduitsByCategorie($idCategorie)
{
    $produits_categ = [];
    try {
        global $connection;
        // Récupérer tous les produits d'une catégorie (y compris
enfants)
        $statement = $connection->prepare("CALL
getProduitsByCategorie(:idCategorie)");
        $statement->bindParam(':idCategorie', $idCategorie,
PDO::PARAM_INT);
        $statement->execute();
        while ($row = $statement->fetch(PDO::FETCH_ASSOC)) {
            $produits_categ[] = $row['idDeclinaison']; // On stocke
tous les id produit de cette catégorie
        }
        $statement->closeCursor();

    } catch (PDOException $e) {
        error_log("Erreur SQL getProduitsByCategorie : " .
$e->getMessage());
    }

    return $produits_categ;
}
?>
```

2. Trace d'exécution

Tri par regroupement

Catalogue

Toutes qualités ▾ Aucun tri Regroupement ▾ Toutes les catégories ▾ Aucun tri de prix ▾

Produit	Qualité	Description	Prix	Actions
Potion de mana majeure	Commun	Potion de mana majeure Potion de mana 19.99€	Voir les Détails Ajouter au panier	
Potion de mana mineure	Commun	Potion de mana mineure Potion de mana 7.99€	Voir les Détails Ajouter au panier	
Potion de Résistance à la foudre	Peu commun	Potion de Résistance à la foudre Potion de résistance 9.99€	Voir les Détails Ajouter au panier	
Potion de Résistance à la glace	Peu commun	Potion de Résistance à la glace Potion de résistance 9.99€	Voir les Détails Ajouter au panier	

Tri par catégorie

Catalogue

Toutes qualités ▾ Aucun tri Regroupement ▾ Armes ▾ Toutes les sous-catégories ▾
Prix croissant ▾

Produit	Qualité	Description	Prix	Actions
Hache de guerre	Commun	Hache de guerre Hache 12.99€	Voir les Détails Ajouter au panier	
Dague de voleur	Commun	Dague de voleur Dague 15.99€	Voir les Détails Ajouter au panier	
Arc long	Commun	Arc long Arme de trait 19.99€	Voir les Détails Ajouter au panier	

Tri par sous-catégorie

Catalogue

Toutes qualités ▾ Aucun tri Regroupement ▾ Armes ▾ Épées ▾
Aucun tri de prix ▾

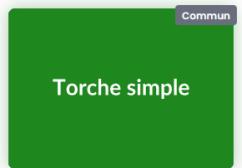
Produit	Qualité	Description	Prix	Actions
Épée courte	Commun	Épée courte Épée 20.49€	Voir les Détails Ajouter au panier	
Épée à deux mains	Commun	Épée à deux mains Épée 34.99€	Voir les Détails Ajouter au panier	
Dragon Slayer	Leg	Dragon Slayer Arme Légendaire 499.99€	Voir les Détails Ajouter au panier	

Tri par prix

Catalogue

Toutes qualités ▾ Aucun tri Regroupement ▾ Toutes les catégories ▾ Prix croissant ▾

Aucun tri de prix
Prix croissant
Prix décroissant

 <p>Peu commun</p> <p>Flèche magique (Unité)</p> <p>Flèches magiques</p> <p>1.99€</p> <p>Voir les Détails</p> <p>Ajouter au panier</p>	 <p>Commun</p> <p>Torche simple</p> <p>Torche simple</p> <p>Source de lumière</p> <p>2.00€</p> <p>Voir les Détails</p> <p>Ajouter au panier</p>	 <p>Commun</p> <p>Ration de voyage</p> <p>Ration de voyage</p> <p>Nourriture</p> <p>5.00€</p> <p>Voir les Détails</p> <p>Ajouter au panier</p>	 <p>Potion de soin mineure</p> <p>Potion de soin</p> <p>7.99€</p> <p>Voir les Détails</p> <p>Ajouter au panier</p>
--	--	--	--

6. Consulter un produit

[US 09]: En tant que visiteur, je veux pouvoir voir le détail d'un produit, avec photos, description, stock, prix, prix soldé, caractéristiques, avis clients, réponses de l'entreprise, produits apparentés et produits aimés par les Visiteurs, afin de me décider.

1. Script

```
<?php
require_once("./connec.inc.php");
//Si on essaye d'accéder à la page trait, on redirige direct vers le catalogue
if (basename($_SERVER['SCRIPT_FILENAME']) === basename(__FILE__)) {
    header('Location: catalogue.php');
    exit;
}

//On récup l'id du produit en get si il existe, sinon null
$idProduit = $_GET['id'] ?? null;

if (!$idProduit || !ctype_digit($idProduit)) { //Si on a pas reçu l'id du produit OU l'id du produit n'est pas un nombre /chiffres
    header("Location: index.php"); //Redirection vers index
    exit;
}

//On essaie de charger le produit
$infos_produit = chargerProduit($idProduit);
if (!$infos_produit) { //Si rien n'a été trouvé sur ce produit (inexistant)
    $erreur = "Produit introuvable.";
    header("Location: catalogue.php?msgErreur=".urlencode($erreur));
    exit;
}

//Chargement des données nécessaires
$stab_varianes = chargerProduitsVariantes($idProduit);
$stab_avis = chargerAvis($idProduit); //Chargement des avis
$stab_produitsSimilaires = chargerProduitsSimilaires($idProduit);

//Stockage des infos du produit
```

```

$nom_produit = $infos_produit['libelleDeclinaison'];
$qualite = $infos_produit['qualite'];
$description = $infos_produit['descDeclinaison'];
$ficheTechnique = $infos_produit['ficheTechnique'];
$prix = $infos_produit['prix'];
$prix_solde = $infos_produit['prixSolde'];
$qte_stock = $infos_produit['qteStock'];
$provenance = $infos_produit['provenance'];
$moyenne = $infos_produit['moyenneAvis'];
$photos_produit = chargerPhotosProduit($idProduit, $nom_produit);
/***
 * Récupère dans la BD toutes les infos sur le produit
 */
function chargerProduit($idProduit)
{
    $tab_produit = [];
    try {
        global $connection;
        $statement = $connection->prepare("SELECT * FROM
DeclinaisonProduit D, Produit P WHERE D.idDeclinaison = ? AND
P.idProduit = D.idProduit"); //Récup les infos sur le produit
        $statement->execute([$idProduit]); //Executer la commande
        $tab_produit = $statement->fetch(PDO::FETCH_ASSOC)); //Récup
un tableau associatif
        return $tab_produit; //On renvoie le tableau des infos du
produit
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerProduit : " . $e->getMessage());
        // retour tab vide si ça plante donc on sera automatiquement
redirigé à index (voir ligne 19)
    }
    return $tab_produit;
}

/***
 * Récupère dans la BD tous les avis
 */
function chargerAvis($idProduit)
{
    $tab_avis = [];
    try {

        global $connection;

```

```

        $statement = $connection->prepare("SELECT DISTINCT * FROM
DeclinaisonProduit as D, Avis as A, Client as C WHERE D.idDeclinaison =
? AND A.idDeclinaisonProduit = D.idDeclinaison AND A.idClient =
C.idClient"); //Récup tous les avis
        $statement->execute([$idProduit]); //Executer la commande
        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
//Récup un tableau associatif
            $tab_avis[] = $ligne;
        }
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerAvis : " . $e->getMessage());
        //en cas d'erreur un tableau vide sera retourné
    }

    return $tab_avis; //On renvoie le tableau des avis du produit
}

/**
 * Récupère dans la BD toutes les réponses de l'avis
 */
function chargerReponsesAvis($idAvis)
{
    $reponses = [];
    try {

        global $connection;
        $statement = $connection->prepare("SELECT * FROM ReponseAvis R,
Administrateur A WHERE R.idAvis = ? AND A.idAdmin = R.idAdmin");
//Récup toutes les réponses
        $statement->execute([$idAvis]); //Executer la commande
        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
//Récup un tableau associatif
            $reponses[] = $ligne;
        }
    } catch (PDOException $e) {
        error_log("Erreur SQL chargerReponseAvis : " .
$e->getMessage());
        //en cas d'erreur un tableau vide sera retourné
    }
}

```

```

        return $reponses; //On renvoie le tableau des réponses

    }

/***
 * Récupère dans la BD tous les produits similaires
 */
function chargerProduitsSimilaires($idProduit)
{
    $tab_produitsSimilaires = [];
    try {

        global $connection;
        //On met d'abord les produits apparentés
        $statement = $connection->prepare("CALL
getDeclinaisonsApparente(:idProduit)"); //Récup tous les produits
apparentés
        $statement->execute(['idProduit' => $idProduit]); //Executer la
commande
        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
//Récup un tableau associatif
            $tab_produitsSimilaires[$ligne['idDeclinaison']] = $ligne;
        }
        $statement->closeCursor();
    } catch (PDOException $e) {

        error_log("Erreur SQL chargerProduitsSimilaires : " .
$e->getMessage());
        //en cas d'erreur un tableau vide sera retourné

    }
    return $tab_produitsSimilaires; //On renvoie le tableau des
produits apparentés

}

/***
 * Fonction permettant de charger les produits "Variantes" d'une
déclinaison
 */
function chargerProduitsVariantes($idDeclinaison)

```

```

{
    $produits = [];

    try {
        global $connection;
        $statement = $connection->prepare("SELECT D2.* FROM
DeclinaisonProduit D1, DeclinaisonProduit D2
        WHERE D1.idDeclinaison = :idDeclinaison1 AND D1.idProduit =
D2.idProduit AND D2.idDeclinaison != D1.idDeclinaison"); //Récup toutes
les variantes
        $statement->execute(['idDeclinaison1' => $idDeclinaison]);
//Executer la commande

        while ($ligne = $statement->fetch(PDO::FETCH_ASSOC)) {
//Récup un tableau associatif
            $produits[] = $ligne;
        }
    } catch (PDOException $e) {

        error_log("Erreur SQL chargerProduitsVariantes : " .
$e->getMessage());

        //en cas d'erreur un tableau vide sera retourné
    }
    return $produits; //On renvoie le tableau des produits
}

/**
 * Fonction permettant d'afficher les étoiles d'une note
 */
function afficherEtoiles($note)
{
    if ($note > 0 && $note != null) {
        //Si la note n'est pas un nombre rond
        $entier = floor($note); //On met la note en entier
        $reste = $note - $entier; //On calcule le reste

        // On affiche toutes les étoiles pleines
        for ($i = 0; $i < $entier; $i++) {
            echo '<span class="text-yellow-500 text-lg">★</span>';
        }

        // demi-étoile si nécessaire (ici on arrondit au supérieur 0.5)
        if ($reste >= 0.5) {

```

```

        echo '<span class="text-yellow-500 text-lg">★</span>';
        $entier++;
    }

    // compléter jusqu'à 5 étoiles
    for ($i = $entier; $i < 5; $i++) {
        echo '<span class="text-gray-300 text-lg">★</span>';
    }
} else {
    // si pas de note, afficher 5 étoiles grises
    for ($i = 0; $i < 5; $i++) {
        echo '<span class="text-gray-300 text-lg">★</span>';
    }
}
}

```

2. Trace d'exécution

Consultation du produit

Potion de mana mineure



Commun
★★★★★ 1 avis

7.99€ Autres versions :

Description
Régénère une petite quantité de mana (20%)

Caractéristiques Techniques
Contenance: 50ml, Effet: Instantané, Poids: 0.1kg

Provenance
Falconia

Quantité :

[Ajouter au panier](#) [Acheter maintenant](#)

Section avis

Avis des Aventuriers

Dubois Marie

2025-12-03 08:55:46



Potion incroyable ! Je recommande

Réponse de LeRoi Merlin 🌟

LeBlanc Gandalf : Merci du retour. Nous sommes enchantés de vous savoir satisfait !

Section produits apparentés

Artefacts Similaires

Peu commun

Potion de Résistance à la foudre

Potion de Résistance à la foudre

Potion de Résistance à la foudre

9.99€

[Voir les Détails](#)

[Ajouter au panier](#)

Peu commun

Potion de Résistance à la glace

Potion de Résistance à la glace

Potion de Résistance à la glace

9.99€

[Voir les Détails](#)

[Ajouter au panier](#)

Peu commun

Potion de Résistance au feu

Potion de Résistance au feu

Potion de Résistance au feu

9.99€

[Voir les Détails](#)

[Ajouter au panier](#)

Peu commun

Potion de Résistance au poison

Potion de Résistance au poison

Potion de Résistance au poison

9.99€

[Voir les Détails](#)

[Ajouter au panier](#)

Co...
Potion de soin majeu...
Potion de soin majeu...
Potion de soin majeu...
19.99€
[Voir les Détails](#)
[Ajouter au panier](#)

7. Ajout d'un produit au panier

[US:10] En tant que client, je veux ajouter un produit affiché dans le panier, afin de préparer ma commande.

1. Script

```
<?php

/**
 * Traiter la requete entrante
 */

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    if (isset($_POST['panier'])) { //Simple ajout au panier

        $id = $_POST["idProduit"] ?? null; //Si l'id produit est
présent on le récup sinon null

        $qte = $_POST["qte"] ?? 1; //Si la qte est présente on la
récup, sinon 1

        $qte = filter_var($qte, FILTER_VALIDATE_INT, ["options" =>
["default" => 1]]); //Conversion en entier

        if ($id && is_int($qte)) { //Si l'id n'est pas null est que
c'est bien un entier

            ajoutPanier($id, $qte); //On ajoute au panier
        }
    }
}
```

```
if (isset($_POST['achat'])) { //Achat direct

    // Redirection vers la page achat

    $previous = $_SERVER['HTTP_REFERER'] ?? 'index.php';

    header("Location: $previous");

    exit;

}

//Après ajout au panier / ou échec, on redirige vers la page
précédente

$previous = $_SERVER['HTTP_REFERER'] ?? 'index.php';

header("Location: $previous");

exit;

}

/**

 * Permet d'ajouter un produit (avec sa qte) dans le panier

 */

function ajoutPanier($idProduit, $qte)

{

    $panier = [];

    //On récup le panier déjà existant si il existe

    if (isset($_COOKIE['panier'])) {

        $panier = json_decode($_COOKIE['panier'], true); //On récup le
panier en un tableau associatif

        if (!array($panier)) { //Si le panier est corrompu / incorrect

            $panier = [];

        }

    }

}
```

```
//Si le produit est déjà présent de le panier : mise à jour de la
qte

if (isset($panier[$idProduit])) {

    $panier[$idProduit] += $qte; //Mise à jour

} else { //Sinon on enregistre le produit avec sa quantité

    $panier[$idProduit] = $qte;

}

//Quantité max

if ($panier[$idProduit] > 99) {

    $panier[$idProduit] = 99;

}

//Si la quantité à baissé en dessous de 0 on suppr le produit

if ($panier[$idProduit] <= 0) {

    unset($panier[$idProduit]);

}

//Mettre à jour le cookie (on enregistre en json), 14 jours

setcookie('panier', json_encode($panier), time() + (14 * 24 * 60 *
60), '/');

}
```

2. Trace d'exécution

Sélection du bouton "Ajouter au panier"

Depuis la page de consultation du produit :

Potion de mana mineure

Commun
★★★★★ 1 avis

7.99€ Autres versions :

Description
Régénère une petite quantité de mana (20%)

Caractéristiques Techniques
Contenance: 50ml, Effet: Instantané, Poids: 0.1kg

Provenance
Falconia

Quantité :

Ajouter au panier **Acheter maintenant**

Depuis le catalogue :

Catalogue

Toutes qualités Aucun tri Regroupement Alchimie & Potions Toutes les sous-catégories
Aucun tri de prix

Commun	Commun	Peu commun	Peu commun
Potion de mana majeure	Potion de mana mineure	Potion de Résistance à la foudre	Potion de Résistance à la glace
Potion de mana majeure Potion de mana 19.99€	Potion de mana mineure Potion de mana 7.99€	Potion de Résistance à la foudre Potion de résistance 9.99€	Potion de Résistance à la glace Potion de résistance 9.99€
Voir les Détails	Voir les Détails	Voir les Détails	Voir les Détails

Sélection du bouton "Mise à jour du logo panier"



8. Consulter le panier

[US:11] En tant que client, je veux visualiser mon panier, modifier la quantité d'un produit et supprimer un produit, afin de gérer mes achats.

1. Script

```
<?php

// ----

// Lecture du cookie

// ----

$panier = [];

if (isset($_COOKIE['panier'])) {

    $panier = json_decode($_COOKIE['panier'], true);

    if (!is_array($panier)) {

        $panier = [];

    }

}

// ----

// Récupération produits depuis la BDD

// ----

$panierDetails = [];

if (!empty($panier)) {

    foreach ($panier as $idDeclinaison => $qte) {

        $stmt = $connection->prepare("

            SELECT
```

```
        dp.idDeclinaison,
        dp.prix,
        dp.prixSolde,
        dp.libelleDeclinaison,
        p.libelle

    FROM DeclinaisonProduit dp

    INNER JOIN Produit p ON p.idProduit = dp.idProduit

    WHERE dp.idDeclinaison = ?

    ");

$stmt->execute([$idDeclinaison]);

$produit = $stmt->fetch(PDO::FETCH_ASSOC);

if ($produit) {

    $prix = $produit['prixSolde'] ?? $produit['prix'];

    $panierDetails[$idDeclinaison] = [
        'nom'      => $produit['libelle'] . ' - ' .
$produit['libelleDeclinaison'],
        'prix'      => $prix,
        'quantite' => $qte
    ];
}

}

}

// -----
// Calculs
// -----
```

```

$sousTotal = 0;

foreach ($panierDetails as $item) {

    $sousTotal += $item['prix'] * $item['quantite'];

}

$reduction = 0;

$total = $sousTotal - $reduction;

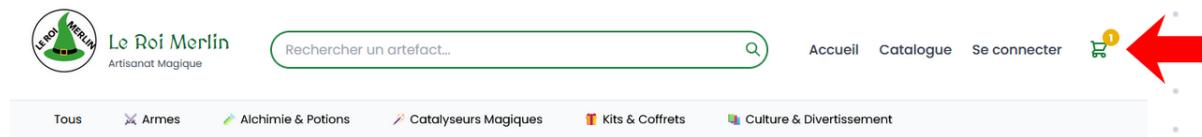
?>

```

2. Trace d'exécution

Après avoir ajouté un produit au panier comme vu [ici](#), on peut consulter le contenu du panier.

Sélection de l'icône panier :



Contenu du panier :

Résumé	
Sous-total	7,99 €
Réduction	- 0,00 €
Total	7,99 €

Passer à la Caisse

[Continuer mes achats](#)

Possibilité d'augmenter / diminuer la quantité :

Votre Panier



Potion de mana - Potion de mana mineure

6 x 7,99 €



47,94 €



Aucun produit présent dans le panier :

Votre Panier

Votre panier est vide

[Continuer mes achats](#)

Conclusion

Ce livrable présente les scripts PHP et les traces d'exécution des user stories prioritaires du projet. Les fonctionnalités développées couvrent la gestion des comptes utilisateurs (création, connexion et déconnexion) ainsi que le système de recherche et de filtrage des produits par différents critères.

Les scripts intègrent les bonnes pratiques de sécurité : hachage des mots de passe, validation des données, utilisation de requêtes préparées pour prévenir les injections SQL, et gestion sécurisée des sessions. Le système de recherche permet de combiner plusieurs filtres (mot-clé, catégorie, regroupement, qualité) et d'effectuer des tris par prix.

Les traces d'exécution démontrent le bon fonctionnement des fonctionnalités dans divers scénarios, y compris la gestion des erreurs. L'ensemble répond aux besoins définis et constitue une base fonctionnelle pour la poursuite du projet.