

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS**SESSION 2023****Épreuve E5 - Conception et développement d'applications (option SLAM)****ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)**

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 2
Nom, prénom : VAQUIÉ Timothé		N° candidat : 02046968914
<input type="checkbox"/> Épreuve ponctuelle <input checked="" type="checkbox"/> Contrôle en cours de formation		Date : 25/04/2023
Organisation support de la réalisation professionnelle		
Intitulé de la réalisation professionnelle Application d'inventaire d'un restaurant		
Période de réalisation : Mars - Avril 2023 Lieu : Quillan Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
Compétences travaillées <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données 		
Conditions de réalisation¹ (ressources fournies, résultats attendus) <ul style="list-style-type: none"> - Ressources fournies : Prise d'informations sur place, recherche sur internet, expression du contexte, expression des besoins. - Résultats attendus : Application permettant la gestion de l'inventaire d'un restaurant. 		
Description des ressources documentaires, matérielles et logiciels utilisés² <ul style="list-style-type: none"> - Ressources documentaires : Recherches sur le web, documentations officielles de Bootstrap, PHP, JavaScript. - Matérielles et logiciels utilisés : Xampp, Visual Studio Code, Android Studio, StarUML. - Langages utilisés : PHP, CSS, Java, Javascript. - Framework : Bootstrap. 		
Modalités d'accès aux productions³ et à leur documentation⁴ <p>Dépôt Github : https://github.com/WallansDev/projet-2</p>		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.*

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Sommaire

I - Présentation.....	2
1.1) Présentation de la société.....	2
1.2) Contexte.....	2
1.3) Description du contenu et des types utilisateurs.....	2
II - Langages & Logiciels.....	2
2.1) Langages.....	2
2.2) Logiciels.....	3
2.3) Autres.....	3
III - Base de données.....	3
3.1) Informations pour la construction.....	3
3.2) Implémentation en ligne SQL.....	4
3.3) Schémas MERISE.....	4
3.3.1) MCD.....	4
3.3.2) MLD.....	4
3.4) Schéma UML.....	5
3.4.1) DCU.....	5
V - Programmation.....	5
5.1) Inclusions.....	5
5.2) Fonctions.....	7
5.3) Sécurisation des mots de passe.....	7
5.4) Utilisation de Bootstrap.....	8
5.5) Architecture générale.....	9
5.6) POO.....	9
5.7) Changement des mots de passe.....	11
5.8) Réinitialisation des mots de passe.....	11
5.9) Historisation des actions.....	13
VII - L'application.....	14
7.1) Vision globale.....	14
7.1.1) login.php.....	14
7.1.2) change-password.php.....	14
7.2) Vision 'patron'.....	14
7.2.1) index.php.....	14
7.2.2) add-stock.php.....	15
7.2.3) update-stock.php.....	15
7.2.4) delete-stock.php.....	15
7.2.5) account.php.....	15
7.2.6) user-add.php.....	16
7.2.7) user-update.php.....	16
7.2.8) user-delete.php.....	16
7.2.9) reset-password.php.....	16
7.3) Vision 'serveur'.....	17
7.3.1) index.php.....	17

7.3.2) update-stock.php.....	17
7.3.3) account.php.....	17
VIII - Possibilités d'ajout.....	18

I - Présentation

1.1) Présentation de la société

Le Terminus est un bar/restaurant, se situant au 45 boulevard Charles de Gaulle dans la commune de Quillan faisant partie de la région de l'Occitanie. Il se démarque des autres entreprises du milieu de la restauration par sa capacité à s'adapter au cours du temps, par la proposition de plats divers et variés et un large temps de travail (7j/7).

L'entreprise dispose d'une terrasse devant le bar et une autre dans le parc servant à la restauration. Situé devant la gare de Quillan, le restaurant attire touristes et voyageurs.

1.2) Contexte

Depuis 2016, Yannick Castel en est le principal dirigeant, ce dernier m'a contacté afin que je réalise une application permettant la gestion de l'inventaire de son restaurant et qui serait disponible sur téléphone.

1.3) Description du contenu et des types utilisateurs

- Serveur:

- *Rechercher un produit.*
- *Modification du mot de passe du compte.*
- *Modifier uniquement la quantité d'un produit.*

- Patron:

- *Hérite des mêmes accès que 'secrétaire'.*
- *Réinitialiser le mot de passe de tous les comptes.*
- *Regarder les logs.*

- Administrateur :

- *Tous les accès.*

II - Langages & Logiciels

2.1) Langages

- HTML 5, CSS, JavaScript, pour représenter les pages web.
- PHP 8.1 (POO), pour faire des pages web dynamiques.
- SQL, pour la base de données.

2.2) Logiciels

- Visual Studio Code, IDE général.
- StarUML, Construction des schémas relationnels.
- Xampp, logiciel pour mettre en place un serveur local.
- Android Studio, IDE pour développement Android.

2.3) Autres

- Bootstrap, framework.

III - Base de données

3.1) Informations pour la construction

Pour réaliser la base de données, je me suis rendu sur place afin de demander des informations et j'ai pu faire des recherches sur internet. Ces informations m'ont permis de comprendre le fonctionnement d'un inventaire de restaurant.

- On identifie un produit du stock par son *libellé*, sa *quantité* et sa *marque*
- Chaque utilisateur sera reconnu et différencié par un *nom d'utilisateur*, un *poste* et un *mot de passe*.

Base de données :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
stock	Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
users	Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
2 tables Somme		15	InnoDB	utf8mb4_0900_ai_ci	48,0 kio	0 o

Table 'stock':

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	IdStock	bigint		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT
2	LibelleStock	varchar(50)	utf8mb4_unicode_ci		Oui	NULL		
3	MarqueStock	varchar(25)	utf8mb4_unicode_ci		Oui	NULL		
4	QteStock	int			Oui	NULL		

Table 'users':

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	IdUser	bigint		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT
2	Username	varchar(30)	utf8mb4_unicode_ci		Non	Aucun(e)		
3	JobUser	varchar(20)	utf8mb4_unicode_ci		Non	Aucun(e)		
4	PasswordHash	varchar(64)	utf8mb4_unicode_ci		Non	Aucun(e)		

3.2) Implémentation en ligne SQL

- Table 'stock'

```

33  CREATE TABLE `stock` (
34    `IdStock` bigint UNSIGNED NOT NULL,
35    `LibelleStock` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
36    `MarqueStock` varchar(25) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
37    `QteStock` int DEFAULT NULL,
38  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
39

```

- Table 'users'

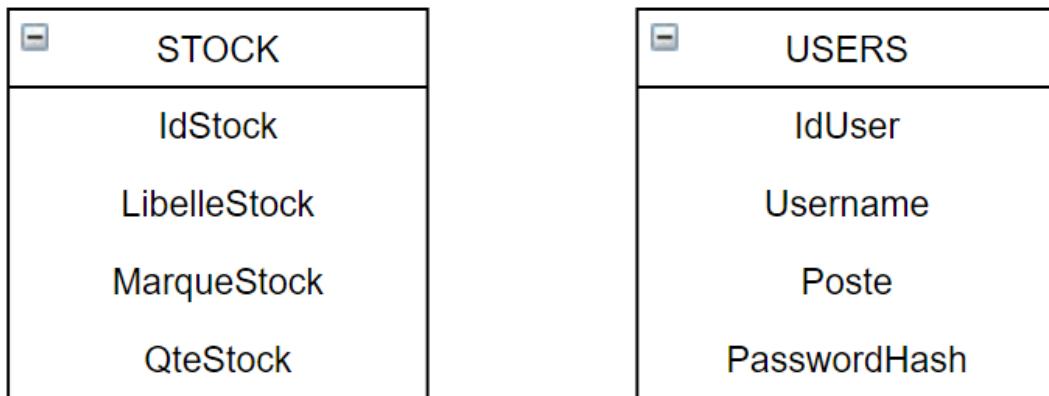
```

61  CREATE TABLE `users` (
62    `IdUser` bigint UNSIGNED NOT NULL,
63    `Username` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
64    `JobUser` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL,
65    `PasswordHash` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
66  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
67

```

3.3) Schémas MERISE

3.3.1) MCD



3.3.2) MLD

STOCK : IdStock, LibelleStock, MarqueStock, QteStock

USERS : IdUser, Username, Poste, PasswordHash

Texte souligné = clé primaire | # = clé étrangère | texte vert = contrainte unique

On décide que LibelleStock et Username soient uniques pour nous permettre facilement de les identifier.
Ex : Il n'existera qu'un seul produit appelé "Chips barbecue 500g" pour éviter les doublons.

3.4) Schéma UML

3.4.1) DCU

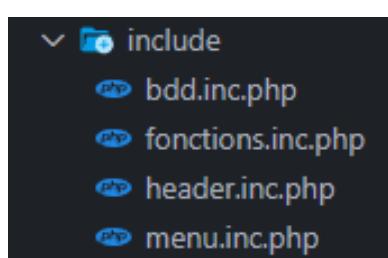


V - Programmation

5.1) Inclusions

On y retrouve tous les fichiers qui vont être inclus dans d'autres fichiers .php
Pour les différencier on met en extension .inc.php.

Dossier include



Il faut différencier include de require. Require interrompt l'exécution du script si le fichier inclus ne peut pas être trouvé ou a des erreurs, tandis qu'include génère une erreur et continue l'exécution du script.

Les inclusions permettent d'insérer le contenu d'un fichier dans un autre fichier.

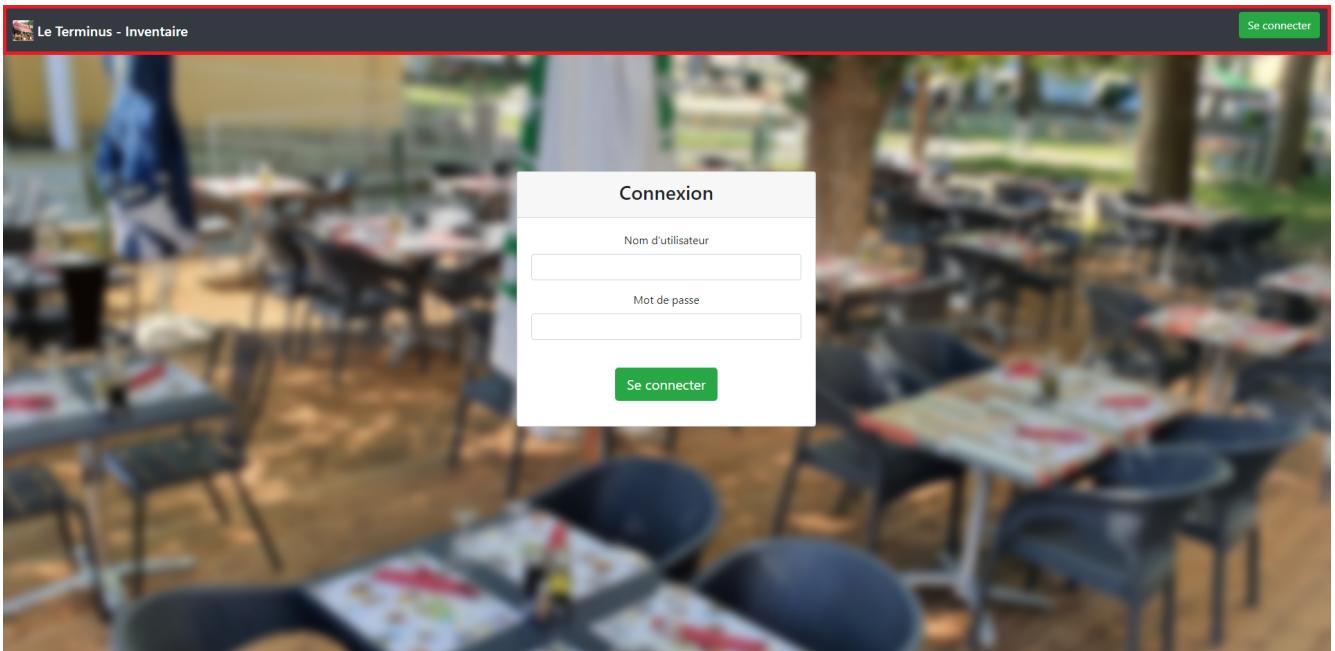
Les exemples suivants vont montrer l'impact que peut avoir l'utilisation d'une inclusion.

Il existe deux modes d'exécution pour une application, à savoir le **mode développement** et le **mode de production**.

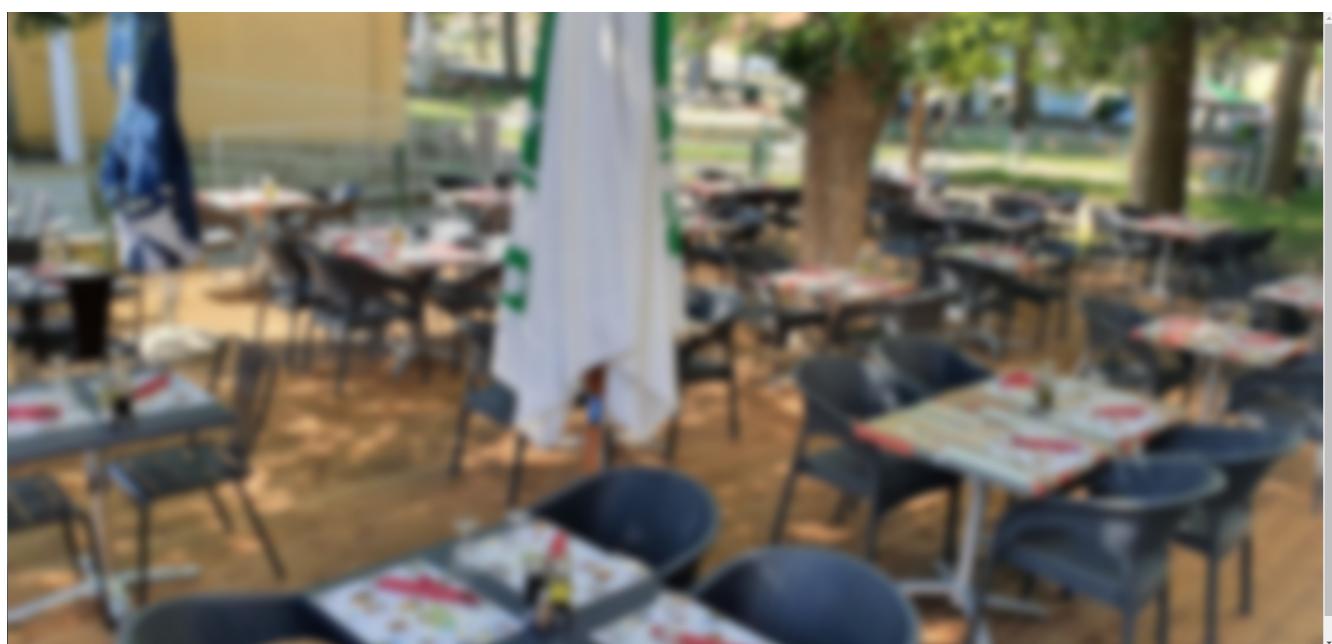
- Le **mode développement** va afficher des erreurs à l'écran (pouvant être dangereux).
- Le **mode de production** préfère ne rien afficher et ainsi informer au minimum un utilisateur standard des erreurs potentielles.

Le choix entre ces deux modes peut être effectué dans le fichier de configuration *php.ini*

Utilisation de '*header.inc.php*' réussie :



Utilisation de '*header.inc.php*' ratée :



5.2) Fonctions

- ``**alertView()**``, cette fonction permet d'afficher une alerte d'information. Celle-ci comporte en entrée une couleur (danger, info, success ... voir doc Bootstrap), un message.

Les autres arguments de la fonction, \$link et \$time, sont facultatifs et permettent de rediriger vers une page après un certain temps d'attente.

```

29   function alertView($couleur, $message, $link, $time)
30   {
31     echo ("<div class='alert alert-$couleur' role='alert'>$message</div>");
32     if (strlen($link) > 1 && strlen($time) > 1) {
33       echo ("<script>setTimeout(function() { window.location.href = './$link';}, $time);</script>");
34     }
35   }

```

- ``**Logs()**``, cette fonction prend en entrée un message qui sera ajouté en première ligne dans le fichier logs.log (fichier servant à l'historisation des actions effectuées sur l'application).

```

18   function Logs($message)
19   {
20     $date = new DateTime();
21     $file = "logs.log";
22     $heure = $date->format("h:i:s");
23     $date = $date->format("y:m:d");
24     $timestamp = strtotime($date);
25     $date = date('d/m/Y', $timestamp);
26
27     $text = file($file, FILE_IGNORE_NEW_LINES);
28
29     array_unshift($text, "[" . $date . " " . $heure . "] " . $message . "\n");
30
31     file_put_contents($file, implode(PHP_EOL, $text) . PHP_EOL);
32   }

```

5.3) Sécurisation des mots de passe

D'après le R.G.P.D.¹, le stockage des mots de passe en clair dans une base de données est interdit.

Il est recommandé de stocker les mots de passe sous forme de hachage (hash) pour les protéger contre les accès non autorisés. De plus, il doivent respecter un certain format (min. 1 caractère spécial + 1 majuscule + 1 minuscule et au moins 8 caractères).

Cette méthode permet d'obtenir une chaîne de caractères de longueur fixe en sortie à partir d'une donnée en entrée. Il est de plus impossible de remonter jusqu'à la donnée initiale.

Les mots de passe enregistrés dans la base de données sont tous hachés avec la méthode `hash()`, en utilisant l'algorithme de hachage `SHA256`.

Pour la vérification d'une connexion j'utilise la fonction `hash()`, native de PHP, où l'on rentre en entrée le nom de l'algorithme utilisé, ici `SHA256`, et le texte à hacher, ici le mot de passe. On vérifie ensuite si le texte haché est égal à celui stocké dans la base de données.

R.G.P.D.¹ : Règlement Général sur la Protection des Données est une réglementation européenne qui vise à renforcer la protection des données personnelles des citoyens de l'Union européenne.

Extrait de login.php :

```

37     $passwordHash = hash("sha256", $password);
38
39     try {
40         $addUser = new User(['Username' => $username, 'PasswordHash' => $passwordHash]);
41         $user = $userManager->verifConnexion($addUser);
42         if ($user) {
43             $_SESSION['username'] = $user->getUsername();
44             $_SESSION['job'] = $user->getJobUser();
45             alertBox("success", "Connexion réussie.", "index.php", 1500);
46             Logs("💡 L'utilisateur '" . $_SESSION['username'] . "' (" . $_SERVER['REMOTE_ADDR'] . ") s'est connecté");
47         } else {
48             alertBox("danger", "Mauvais nom d'utilisateur ou mot de passe.", "", "");
49             Logs("⚠ Un utilisateur (" . $_SERVER['REMOTE_ADDR'] . ") n'a pas réussi à se connecter");
50         }
51     } catch (PDOException $e) {
52         alertBox("danger", "La requête n'a pas pu aboutir.", "", "");
53         Logs("⚠ Un utilisateur (" . $_SERVER['REMOTE_ADDR'] . ") n'a pas réussi à se connecter");
54     }

```

Extrait de userManager.class.php :

```

57     public function verifConnexion(User $user)
58     {
59         $q = $this->_db->prepare("SELECT Username, JobUser, PasswordHash FROM users WHERE Username = :Username AND PasswordHash = :PasswordHash");
60         $q->bindValue(':Username', $user->getUsername());
61         $q->bindValue(':PasswordHash', $user->getPasswordHash());
62         $q->execute();
63
64         $userInfo = $q->fetch(PDO::FETCH_ASSOC);
65         if ($userInfo) {
66             return new User($userInfo);
67         } else {
68             return 0;
69         }
70     }

```

On vient utiliser la méthode `verifConnexion()` de la classe `userManager.class.php` pour vérifier s'il existe dans la table `users` un utilisateur ayant le nom d'utilisateur et le mot de passe haché correspondant aux informations rentrées dans le formulaire de connexion.

Table 'users' de la base de données :

	<input type="button" value="←"/> <input type="button" value="→"/>		<input type="button" value="▼"/>	IdUser	Username	JobUser	PasswordHash
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	1	administrateur	admin	00fcddde26dd77af7858a52e3913e6f3330a32b3121a61bce91...
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	2	Yannick	patron	00fcddde26dd77af7858a52e3913e6f3330a32b3121a61bce91...
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	3	Patrice	patron	00fcddde26dd77af7858a52e3913e6f3330a32b3121a61bce91...
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	4	Pierrick	serveur	00fcddde26dd77af7858a52e3913e6f3330a32b3121a61bce91...
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	5	Pierre	serveur	540eda905bdfc262fbae050565c539265cd0a92391d5d2838a...
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	8	Michel	patron	00fcddde26dd77af7858a52e3913e6f3330a32b3121a61bce91...

5.4) Utilisation de Bootstrap

J'ai utilisé le framework front-end Bootstrap, qui a facilité la création de l'application et son design.

Ajout de Bootstrap :

```

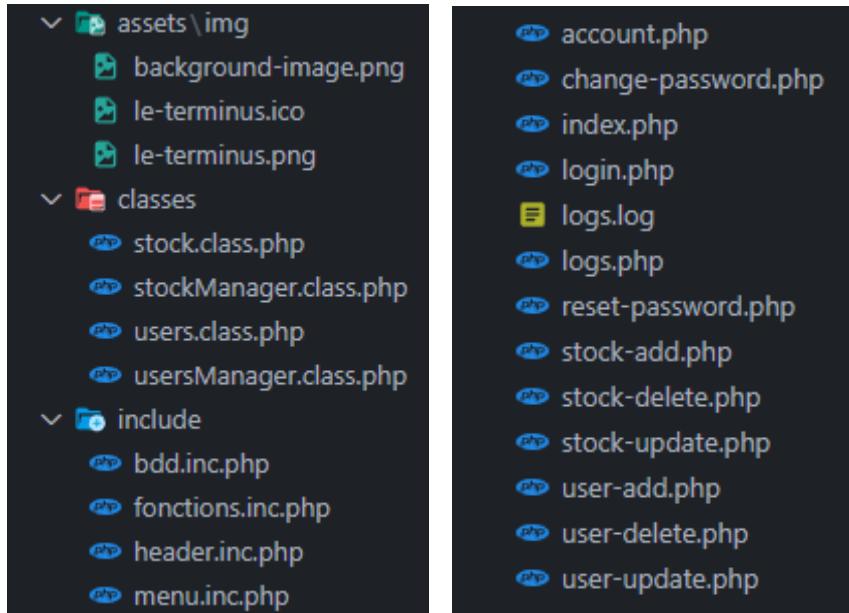
1   <head>
2     <meta charset="utf-8">
3     <meta name="viewport" content="width=device-width, initial-scale=1">
4     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
5       integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

```

```

61 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzbzo5smXKp4YFRvH+8abtTE1Pi6jizo" crossorigin="anonymous">
62 </script>
63 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01cIHTMGa3JDzwrnQq4sF86dIHNDz8W1"
64 crossorigin="anonymous">
65 </script>
66 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
67 crossorigin="anonymous">
68 </script>
69 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
70 integrity="sha384-u1OkNcvxWvY5kfmNBILK2hRnQC3Pr17a+RiT6rIHI7NnikvbZlHgTP00mMi466C8" crossorigin="anonymous">
71 </script>
```

5.5) Architecture générale



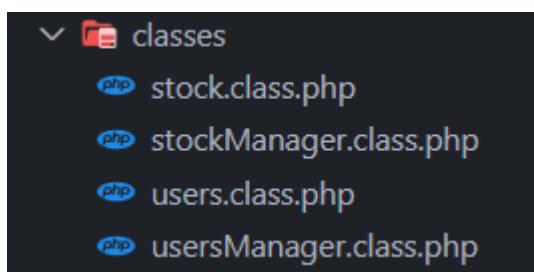
5.6) POO

Chaque classe correspond à une table SQL.

Le dossier classes contient tous les fichiers de classes PHP, chaque classe dispose de 2 fichiers *nom_de_classe.class.php* et *nom_de_classeManager.class.php*.

Le premier encapsule les propriétés et les méthodes nécessaires pour récupérer les données de la table.

Le manager est une classe qui est responsable de la gestion. Elle contient des méthodes pour effectuer des opérations, on y retrouve souvent le C.R.U.D¹.



¹ C.R.U.D. : Create Read Update Delete, acronyme qui désigne les quatre opérations de base qui peuvent être effectuées sur une base de données

- Extrait de la classe 'StockManager.class.php'

```

1 <?php
2 class StockManager
3 {
4     private $_db;
5
6     public function __construct($db)
7     {
8         $this->setDB($db);
9     }
10
11    // ADD
12    public function addStock(Stock $stock)
13    {
14        $q = $this->_db->prepare("INSERT INTO stock (LibelleStock, MarqueStock, QteStock) VALUES (:LibelleStock, :MarqueStock, :QteStock)");
15        $q->bindValue(':LibelleStock', $stock->getLibelleStock());
16        $q->bindValue(':MarqueStock', $stock->getMarqueStock());
17        $q->bindValue(':QteStock', $stock->getQteStock());
18        $q->execute();
19        $id = $this->_db->lastInsertId();
20        return $id;
21    }
22
23    // UPDATE
24    public function updateStock(Stock $stock)
25    {
26        $q = $this->_db->prepare("UPDATE stock SET `LibelleStock` = :LibelleStock, `MarqueStock` = :MarqueStock, `QteStock` = :QteStock WHERE `IdStock` = :IdStock");
27        $q->bindValue(':IdStock', $stock->getIdStock());
28        $q->bindValue(':LibelleStock', $stock->getLibelleStock());
29        $q->bindValue(':MarqueStock', $stock->getMarqueStock());
30        $q->bindValue(':QteStock', $stock->getQteStock());
31        $q->execute();
32    }
33
34    // DELETE
35    public function deleteStock(Stock $stock)
36    {
37        $q = $this->_db->prepare("DELETE FROM stock WHERE `IdStock` = :IdStock");
38        $q->bindValue(':IdStock', $stock->getIdStock());
39        $q->execute();
40    }
41
42    public function getInfoById($id)
43    {
44        $id = (int) $id;
45        $req = $this->_db->query("SELECT IdStock, LibelleStock, MarqueStock, QteStock FROM stock WHERE IdStock = $id");
46        $donnees = $req->fetch(PDO::FETCH_ASSOC);
47
48        return new Stock($donnees);
49    }
50

```

- Extrait de la classe 'Stock.class.php'

```

1 <?php
2 class Stock
3 {
4     // Attributs
5     private $_IdStock;
6     private $_LibelleStock;
7     private $_MarqueStock;
8     private $_QteStock;
9
10    public function __construct(array $donnees)
11    {
12        $this->hydrate($donnees);
13    }
14
15    public function hydrate(array $donnees)
16    {
17        foreach ($donnees as $key => $value) {
18            $method = 'set' . ucfirst($key);
19
20            if (method_exists($this, $method)) {
21                $this->$method($value);
22            }
23        }
24    }

```

```

26    // Getters
27    public function getIdStock()
28    {
29        return $this->_IdStock;
30    }
31
32    public function getLibelleStock()
33    {
34        return $this->_LibelleStock;
35    }
36
37    public function getMarqueStock()
38    {
39        return $this->_MarqueStock;
40    }
41
42    public function getQteStock()
43    {
44        return $this->_QteStock;
45    }
46
47    // Setters
48    public function setIdStock($idStock)
49    {
50        $this->_IdStock = $idStock;

```

5.7) Changement des mots de passe

Chaque utilisateur a la possibilité de changer son mot de passe, un bouton prévu à cet effet est présent sur la page de leur compte.

account.php :

#	Nom d'utilisateur	Poste	Mot de passe
1	administrateur	Admin	
2	Yannick	Patron	
3	Patrice	Patron	
4	Pierrick	Serveur	
5	Pierre	Serveur	
8	Michel	Patron	

Ce bouton redirige l'utilisateur vers une autre page :

change-password.php, modification réussie :

Modification réussie

Mot de passe

Modifier

change-password.php, modification ratée :

Mot de passe

* Le format du mot de passe n'est pas respecté, il est requis :
- 1 majuscule
- 1 minuscule
- 1 caractère spécial
- 8 caractères

Modifier

5.8) Réinitialisation des mots de passe

Chaque utilisateur ayant comme poste patron et administrateur dispose de la possibilité de réinitialiser les mots de passe des utilisateurs.

Lors de l'appui sur le bouton, le mot de passe du compte sélectionné est réinitialisé à celui par défaut (*P@ssword1*). L'utilisateur pourra par la suite se connecter à son compte et procéder au changement de son mot de passe pour plus de sécurité.

Front-end :

reset.php :

#	Nom d'utilisateur	Poste	Mot de passe
1	administrateur	Admin	
2	Yannick	Patron	
3	Patrice	Patron	
4	Jean	Serveur	
5	Pierre	Serveur	
8	Michel	Patron	

reset-password.php :

You êtes sur le point de réinitialiser :

Jean

serveur

Cette action sera irréversible ?

Réinitialiser

reset-password, modification réussie :

Réinitialisation réussite

You êtes sur le point de réinitialiser :

Jean

serveur

Cette action sera irréversible ?

Réinitialiser

Back-end :

On va créer un objet User et lui indiquer l'attribut *IdUser*, ensuite on va appeler la méthode 'resetPassword()' en lui passant en paramètre l'objet créé.

Extrait de reset-password.php :

```

3   $id = $_GET['id'];

$user = $userManager->getInfoById($id);

if (isset($_POST['submit'])) {
    try {
        $resetPassword = new User(['IdUser' => $id]);
        $userManager->resetPassword($resetPassword);
        alertBox("success", "Réinitialisation réussie", 'account.php', 1000);
        Logs("L'utilisateur '" . $_SESSION['username'] . "' (" . $_SERVER['REMOTE_ADDR'] . ") a réinitialisé le mot de passe du compte '" . $user->getUsername() . "'");
    } catch (PDOException $e) {
        alertBox("danger", "La requête n'a pas pu aboutir", '', '');
        Logs("L'utilisateur '" . $_SESSION['username'] . "' (" . $_SERVER['REMOTE_ADDR'] . ") n'a pas réussi à réinitialiser le mot de passe du compte '" . $user->getUsername() . "'");
    }
}

```

Extrait de userManager.class.php :

```

31     public function resetPassword(User $user)
32     {
33         // Mot de passe initial (P@ssword1): 00fcdde26dd77af7858a52e3913e6f3330a32b3121a61bce915cc6145fc44453
34         $q = $this->_db->prepare("UPDATE users SET `PasswordHash` = '00fcdde26dd77af7858a52e3913e6f3330a32b3121a61bce915cc6145fc44453' WHERE `idUser` = :idUser;");
35         $q->bindValue(':idUser', $user->getIdUser());
36         $q->execute();
37     }

```

La méthode va venir remplacer le mot de passe haché de l'utilisateur par un autre par défaut, '**P@ssword1**', dans la base de données. Ainsi, l'utilisateur pourra se connecter en utilisant le mot de passe par défaut. En revanche, il lui faudra le changer dès qu'il peut pour des questions de sécurité.

5.9) Historisation des actions

Les "logs" (ou journaux en français) sont des fichiers qui enregistrent les événements ou les actions qui se produisent sur un système informatique ou une application.

Pour enregistrer les actions, j'ai utilisé une fonction qui m'a permis à chaque fois que je l'utilise d'ajouter une ligne correspondant aux informations de l'activité.

Extrait de fonctions.inc.php :

```

18     function Logs($message)
19     {
20         $date = new DateTime(); // date actuelle
21         $heure = $date->format("h:i:s"); // $heure donnera l'heure. Ex : 12:37:30
22         $date = $date->format("y:m:d"); // $date donnera la date. Ex : 2023-05-08
23         $file = "logs.log"; // Fichier de destination
24         $timestamp = strtotime($date);
25         $date = date('d/m/Y', $timestamp); //On modifie $date pour qu'il affiche la date au format français. Ex : 08/05/2023 au lieu de 2023-05-08
26
27         $text = file($file, FILE_IGNORE_NEW_LINES); // Récupère le contenu du fichier $file.
28         // FILE_IGNORE_NEW_LINES indique à PHP de ne pas inclure les caractères de nouvelle ligne dans chaque élément du tableau.
29
30         array_unshift($text, "[" . $date . " " . $heure . "] " . $message . "\n"); // Ajoute le nouveau message en haut du tableau $text.
31
32         file_put_contents($file, implode(PHP_EOL, $text) . PHP_EOL); // Ecrit le contenu du tableau $text dans le fichier $file
33         // On concatène PHP_EOL qui est un retour à la ligne et le message $text.
34     }

```

Extrait de stock-add.php, exemple d'utilisation :

```

16     if (isset($_POST['submit'])) {
17         $libelle = $_POST['libelle'];
18         $marque = $_POST['marque'];
19         $qte = $_POST['qte'];
20
21         try {
22             $addStock = new Stock(['LibelleStock' => $libelle, 'MarqueStock' => $marque, 'QteStock' => $qte]);
23             $id = $stockManager->addStock($addStock);
24             alertDialog("success", "L'ajout a été fait.", "index.php", 1500);
25             Logs("[+] L'utilisateur '" . $_SESSION['username'] . "' (" . $_SERVER['REMOTE_ADDR'] . ") a ajouté le produit n°$id");
26         } catch (PDOException $e) {
27             if ($e->getCode() == 23000) {
28                 alertDialog("danger", "Ce libelle existe déjà.", '', '');
29                 Logs("▲ L'utilisateur '" . $_SESSION['username'] . "' (" . $_SERVER['REMOTE_ADDR'] . ") à essayé d'ajouter un nouveau produit, mais le libellé rentré existe déjà");
30             } else {
31                 alertDialog("danger", "La requête n'a pas pu aboutir.", '', '');
32                 Logs("▲ L'utilisateur '" . $_SESSION['username'] . "' (" . $_SERVER['REMOTE_ADDR'] . ") n'a pas réussi à ajouter un nouveau le produit");
33             }
34         }
35     }

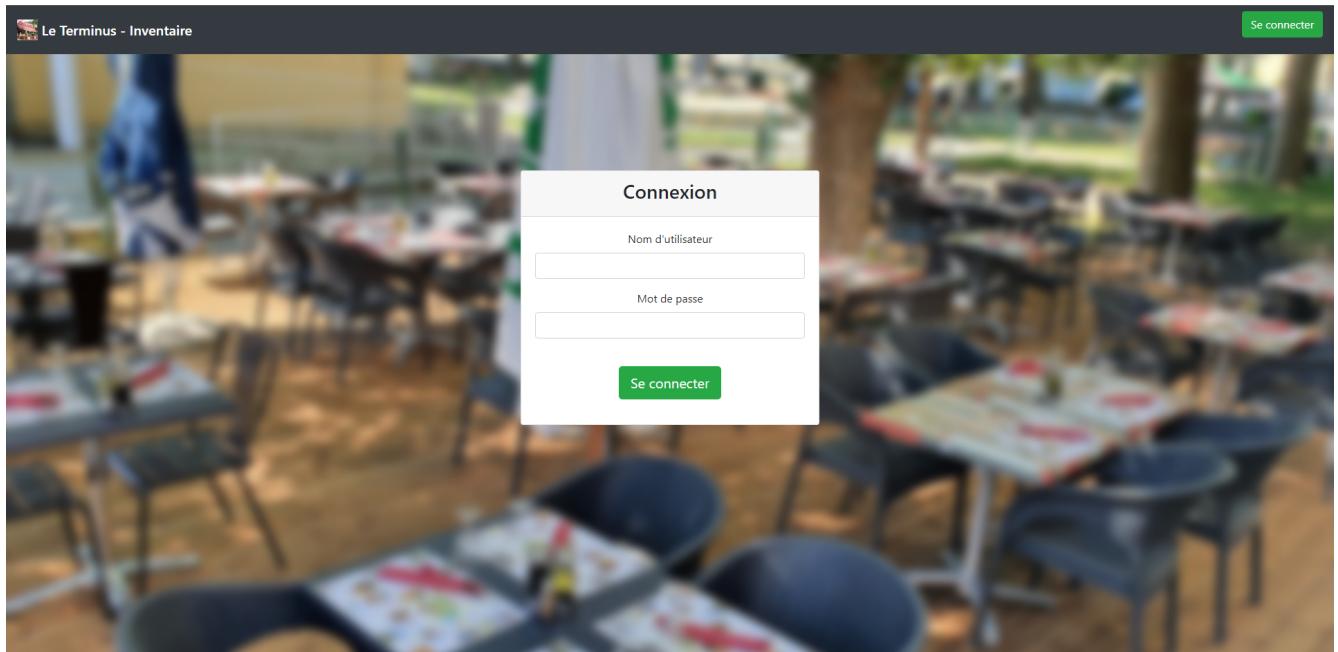
```

VII - L'application

7.1) Vision globale

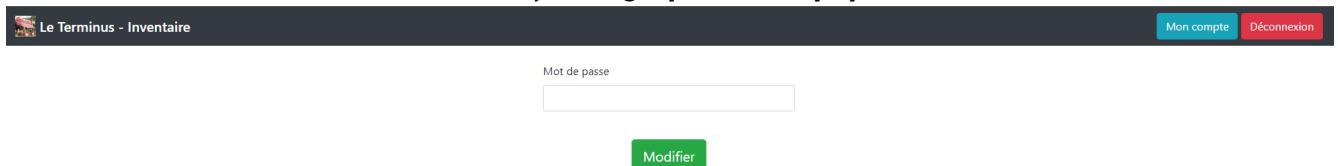
Page de connexion.

7.1.1) login.php



Page de modification du mot de passe.

7.1.2) change-password.php



7.2) Vision 'patron'

Page d'accueil qui montre tous les produits présents dans la base de données.

7.2.1) index.php

#	Libelle	Marque	Quantité
1	Chips Pouket paprika 500g	Lay's	25
2	Chips 250g, goût Barbecue	Lays	4
3	Chips 250g, goût Fromage	Vico	2
4	Cacahuètes 150g	Menguy's	0
5	Jack Daniel's N.7	Jack Daniel's	5
6	Canettes Coca-Cola	Coca-Cola	20
7	Canettes Fanta	Fanta	10
8	Sirop de pêche 1L	Menguy's	2
20	Chips	Lay's	6

Page d'ajout d'un produit dans le stock.

7.2.2) add-stock.php

The screenshot shows a form for adding a product to the stock. It has three input fields: 'Libelle' (Label) containing 'Ex: Chips barbecue 500g', 'Marque' (Brand) containing 'Ex: Lays', and 'Quantité' (Quantity) containing 'Ex: 5'. Below the fields is a green 'Ajouter' (Add) button.

Page de modification d'un produit.

7.2.3) update-stock.php

The screenshot shows a form for modifying a product. It has three input fields: 'Libelle' (Label) containing 'Chips 250g, goût Barbecue', 'Marque' (Brand) containing 'lays', and 'Quantité' (Quantity) containing '4'. Below the fields is a green 'Modifier' (Modify) button.

Page de suppression d'un produit.

7.2.4) delete-stock.php

The screenshot shows a confirmation message: 'Vous êtes sur le point de supprimer :'. It displays the product details: 'Libelle' (Label) 'Chips 250g, goût Barbecue', 'Marque' (Brand) 'lays', and 'Quantité' (Quantity) '4'. Below the message is a red warning: 'Cette action sera irréversible ?' (This action is irreversible?). At the bottom is a red 'Supprimer' (Delete) button.

Page du compte, montre les utilisateurs enregistrés dans la base de données et permet de modifier son mot de passe.

7.2.5) account.php

The screenshot shows a table of registered users. The columns are '#', 'Nom d'utilisateur' (User Name), 'Poste' (Position), and 'Mot de passe' (Password). Each row has a blue edit icon and a red trash icon in the last column. A yellow 'Modifier mon mot de passe' (Change my password) button is located at the top left of the table area.

#	Nom d'utilisateur	Poste	Mot de passe
1	administrateur	Admin	
2	Yannick	Patron	
3	Patrice	Patron	
4	Jean	Serveur	
5	Pierre	Serveur	
6	Michel	Patron	

Page pour ajouter un utilisateur.

7.2.6) user-add.php

The screenshot shows a dark-themed web application header with the title "Le Terminus - Inventaire". Below the header, there are three buttons: "Logs" (blue), "Mon compte" (light blue), and "Déconnexion" (red). The main form area has two input fields: "Nom d'utilisateur" (with placeholder "Ex: Nom d'utilisateur") and "Poste" (with placeholder "... Poste ..."). A green "Ajouter" button is located at the bottom right of the form.

Page de modification d'un utilisateur.

7.2.7) user-update.php

The screenshot shows a dark-themed web application header with the title "Le Terminus - Inventaire". Below the header, there are three buttons: "Logs" (blue), "Mon compte" (light blue), and "Déconnexion" (red). The main form area has two input fields: "Nom d'utilisateur" (containing "Jean") and "Poste" (a dropdown menu showing "serveur" as the selected option, with other choices like "Serveur" and "Patron" visible). A green "Modifier" button is located at the bottom right of the form.

Page de suppression d'un utilisateur.

7.2.8) user-delete.php

The screenshot shows a dark-themed web application header with the title "Le Terminus - Inventaire". Below the header, there are three buttons: "Logs" (blue), "Mon compte" (light blue), and "Déconnexion" (red). The main content area displays a message "Vous êtes sur le point de supprimer :". Below it are two input fields: "Nom d'utilisateur" (containing "Jean") and "Poste" (containing "serveur"). A note "Mot de passe" with a password field containing "#" is also present. A red "Supprimer" button is located at the bottom right.

Vous êtes sur le point de supprimer :

Nom d'utilisateur: Jean Poste: serveur

Mot de passe: #

Cette action sera **irréversible** ?

Supprimer

Page de réinitialisation du mot de passe d'un utilisateur.

7.2.9) reset-password.php

The screenshot shows a dark-themed web application header with the title "Le Terminus - Inventaire". Below the header, there are three buttons: "Logs" (blue), "Mon compte" (light blue), and "Déconnexion" (red). The main content area displays a message "Vous êtes sur le point de réinitialiser :". Below it are two input fields: "Nom d'utilisateur" (containing "Jean") and "Poste" (containing "serveur"). A note "Cette action sera **irréversible** ?" is displayed above a teal "Réinitialiser" button.

Vous êtes sur le point de réinitialiser :

Nom d'utilisateur: Jean Poste: serveur

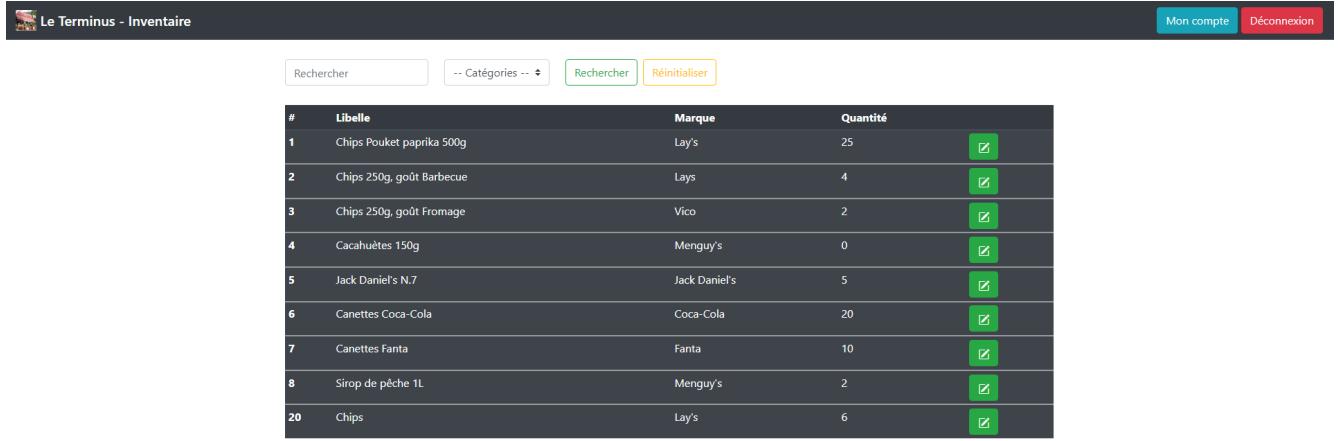
Cette action sera **irréversible** ?

Réinitialiser

7.3) Vision 'serveur'

Page d'accueil qui montre tous les produits présents dans la base de données.

7.3.1) index.php

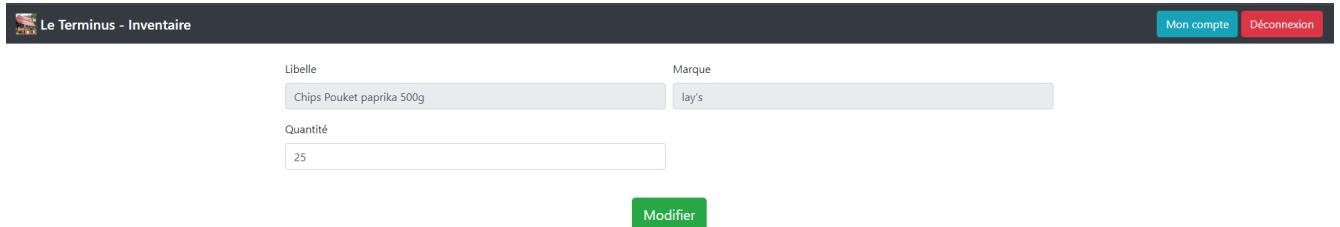


The screenshot shows a table of products with columns: #, Libelle, Marque, Quantité, and a checkbox column. The products listed are:

#	Libelle	Marque	Quantité	
1	Chips Pouket paprika 500g	Lay's	25	<input checked="" type="checkbox"/>
2	Chips 250g, goût Barbecue	Lays	4	<input checked="" type="checkbox"/>
3	Chips 250g, goût Fromage	Vico	2	<input checked="" type="checkbox"/>
4	Cacahuètes 150g	Menguy's	0	<input checked="" type="checkbox"/>
5	Jack Daniel's N.7	Jack Daniel's	5	<input checked="" type="checkbox"/>
6	Canettes Coca-Cola	Coca-Cola	20	<input checked="" type="checkbox"/>
7	Canettes Fanta	Fanta	10	<input checked="" type="checkbox"/>
8	Sirop de pêche 1L	Menguy's	2	<input checked="" type="checkbox"/>
20	Chips	Lay's	6	<input checked="" type="checkbox"/>

Page de modification d'un produit.

7.3.2) update-stock.php



The screenshot shows a form with fields for Libelle, Marque, and Quantité. The values are: Libelle = Chips Pouket paprika 500g, Marque = lay's, and Quantité = 25. A green 'Modifier' button is at the bottom.

Page du compte, permet de modifier son mot de passe.

7.3.3) account.php



The screenshot shows a message: "Vous êtes connecté en tant que Jean". Below it is a yellow button labeled "Modifier mon mot de passe".

VIII - Possibilités d'ajout

Il serait possible dans l'avenir d'ajouter différentes fonctionnalités afin d'améliorer l'application :

- Limiter les tentatives de connexion.
- Alerter lorsque le seuil limite est atteint.