BTS SIO 2022

Support et mise à disposition de services informatiques (E4)

PAGE DE PRÉSENTATION DU DOSSIER

N° de candidat : |0|2|1|4|6|7|1|7|4|7|3|

NOM : DUFOUR

Prénom : Thomas

Date de passage[1] : 24 / 05 /2022	**Heure de passage[1] : 9h30**

CATEGORIE CANDIDAT [2] (UNE CASE A COCHER)	
❑ Scolaire	❑ Ex-scolaire
◪ Apprenti	❑ Ex-apprenti
❑ Formation professionnelle continue	❑ Ex-formation professionnelle continue
❑ Expérience professionnelle 3 ans	

[1] *Informations communiquées sur votre convocation envoyée en mars-avril 2022*

[2] *Informations communiquées sur votre confirmation d'inscription*

Tampon de
l'établissement

BTS SIO – Dossier Etudiant

Justificatif d'acquisition des compétences

Rédacteur(s)	Version	Date	Nb pages
Heilig Fabian	1.1	15/03/2022	10

Epreuve E4

Support et mise à disposition de services informatiques

SOMMAIRE

1 Introduction

L'objectif de ce document est de vous présenter les missions professionnelles que j'ai effectué dans le cadre de ma formation BTS SIO à l'école IRIS de Strasbourg.

Ces missions peuvent être de trois types :

- Effectuées en entreprise durant une alternance
- Effectuées en stage en entreprise
- Effectuées à l'école (compte-rendu de TP, projet collaboratif)

Le type de la mission sera précisé dans chaque cahier des charges.

Ce document se compose des parties suivantes :

Chapitres	Contenu
Chapitres 2 à 8	Présentation des missions, avec pour chacune : - Le cahier des charges - La solution proposée - La gestion de projet - La mise en œuvre - Le bilan du projet

2 Mission 6 : Création d'un site e-commerce de vélo

2.1 Cahier des charges

Type de mission
Mission effectuée à l'école
Contexte
Une société de vente de vélo fait un appel d'offre pour la conception d'un site e-commerce. Notre groupe de prestataire décide de prendre en charge la conception de ce site en se départageant les tâches afin d'être le plus efficient possibles. Dans ce projet, je suis chargé de la conception de la page de connexion et inscription.
Demande du client
- Créer un formulaire de connexion - Créer un formulaire d'inscription - Sécuriser les requêtes pour l'intégration en base de données
Budget disponible
Aucun budget nécessaire pour cette tâche
Outils disponibles
- L'IDE PHP Storm - Un repository GitHub - Git - Une Base de données
Contraintes
La contrainte principale de ce projet fût d'appréhender correctement le travail en groupe sous forme de projet. Il a été difficile pour nous de définir une méthode de travail efficace afin de répondre au besoin de travailler en groupe au sein d'un même projet.

2.2 Etude et conception de la solution

2.2.1 La gestion de projet

Au sein de ce projet, il a fallu définir des méthodes de travail afin d'être dans les temps. Après quelques temps de réception nous avons souhaités partir sur un système de répartition de micro-tâches. Chaque semaine, nous définissons les tâches à effectuer selon des priorités pour ensuite les effectuer jusqu'à la réunion de la semaine suivante.

Pour centraliser nos créations, nous avons créé un répository GitHub ou nous faisons des push chaque semaine.

2.2.1.1 Git et GitHub

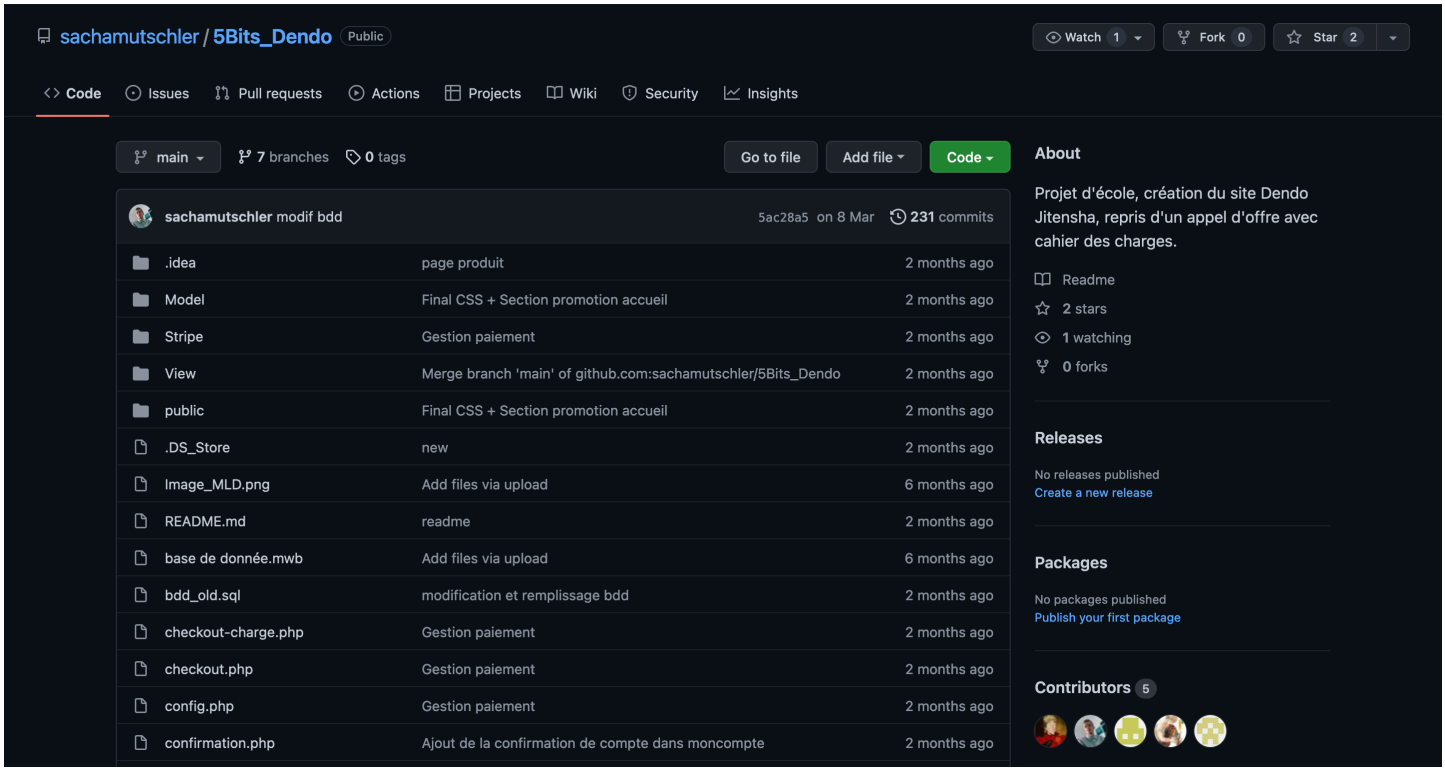
Git est un logicielle de versioning permettant de garder un historique des modifications effectuées sur un projet afin de pouvoir rapidement identifier les changements apportés sur un projet et de revenir à une ancienne version en cas de problèmes.

Aujourd'hui, les logicielles de versioning sont incontournables dans la gestion de grands projets et permettent une approche plus sereine dans la gestion de projet. En effet, si on remarque quelque chose qui ne s'est pas bien passé lors d'un push (publier des changements locaux et les charger vers un dépôt centralisé), on peut toujours revenir sur une ancienne version afin de revenir à une version plus stable.

GitHub est un site web avec un service cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu'à suivre, contrôler et valider les modifications qui sont apportés à un projet. C'est une société offre un service d'hébergement de référentiel Git basé sur le cloud.

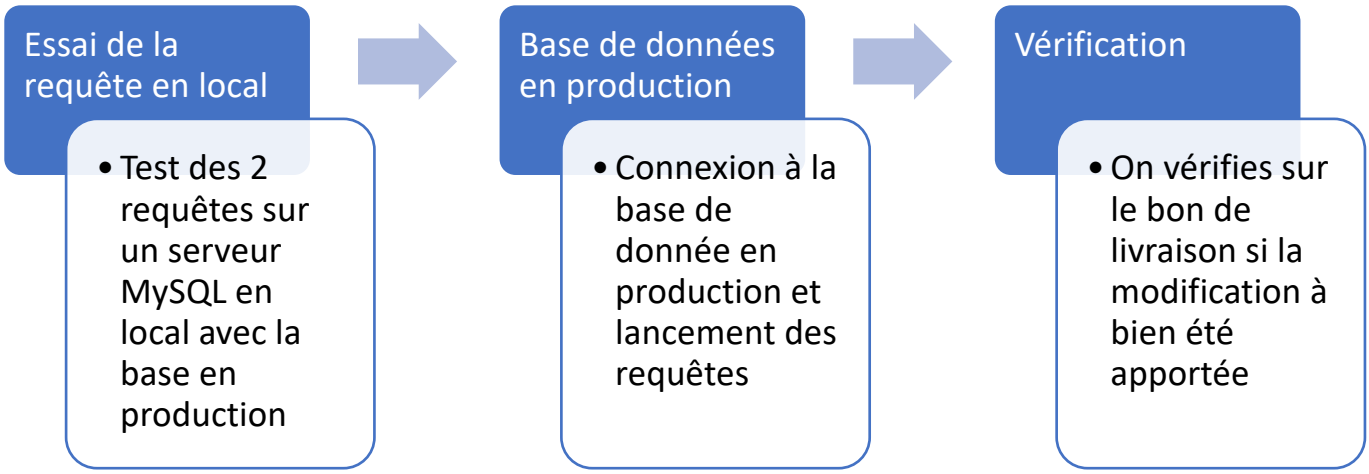
N'importe qui peut s'inscrire et héberger gratuitement un dépôt de code public, ce qui rend ce site web particulièrement populaire auprès de projets open source.

L'utilité de GitHub dans le projet Dendo a été de centraliser le code source de notre groupe au sein d'un répertoire commun afin de mieux appréhender les potentielles erreurs commises lors de modification ou ajout de fonctionnalités. Vous pouvez voir ici, notre repository GitHub :



2.3 Gestion de projet

2.3.1 Planing de déploiement de la solution



2.3.2 Budget

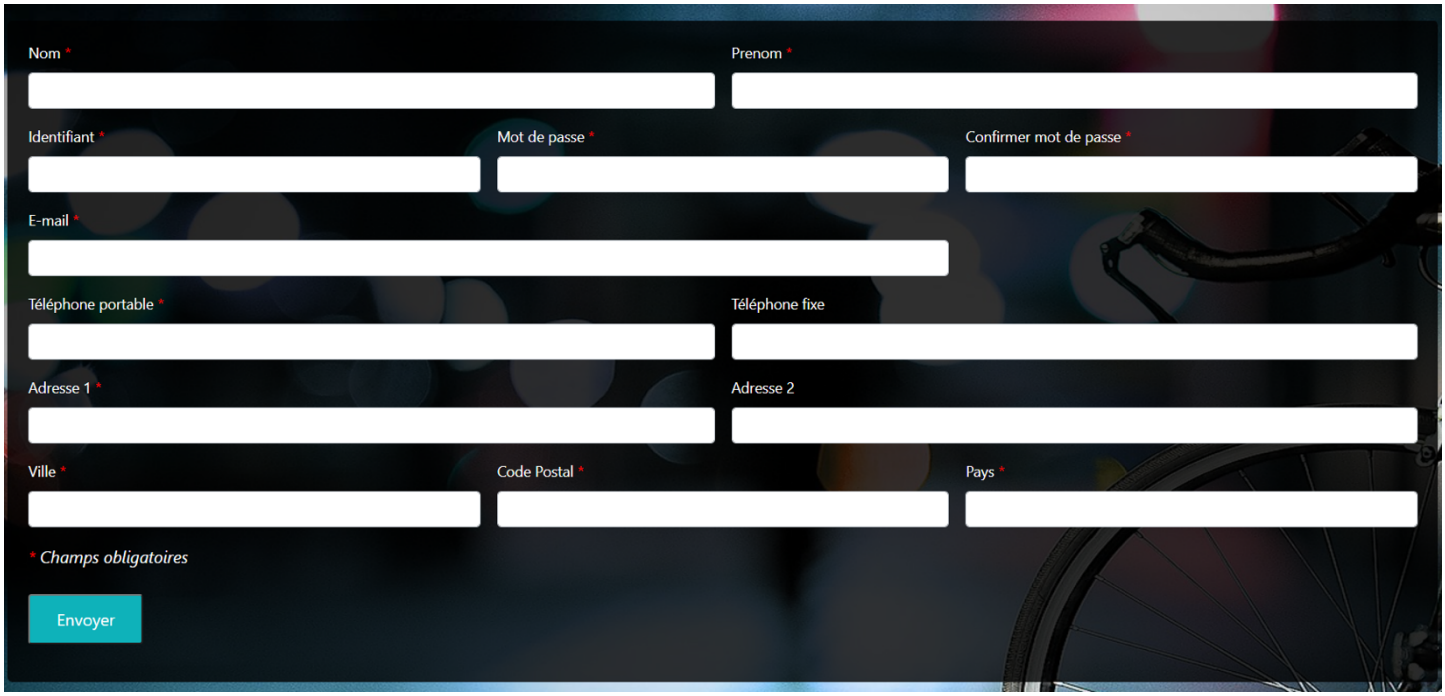
Pas de budget alloué pour cette opération.

2.4 Création d'un formulaire de connexion/inscription

Durant la création de ce projet, il a fallu choisir entre d'abord développer le formulaire d'inscription ou le formulaire de connexion. Le plus logique étant le formulaire d'inscription, c'est par celui-ci que j'ai commencé.

I- Création du formulaire d'inscription

D'après une de nos réunions précédentes, il a été défini les champs nécessaires à la création d'un compte sur le site web. Voici ces champs :



Lorsqu'un utilisateur clique sur le bouton « envoyer » Tout un tas de vérifications sont effectués, si elles ne sont pas respectées, on entre un message d'erreur dans un tableau d'erreur nommé $errors :

```php
$errors = [];
```

Pour en citer quelques-unes, nous pouvons au préalable vérifier si le nom d'utilisateur renseigné est déjà présent dans la base de données, ce qui empêcherait l'inscription :

```php
if  (isset($_POST['identifiant']) && !empty($_POST['identifiant'])){
    $selectExistInBdd = $conn->prepare('SELECT * FROM compte_client WHERE identifiant = :identifiant');
    $selectExistInBdd->bindValue('identifiant', $_POST['identifiant']);
    $selectExistInBdd->execute();
    $selectExistInBdd = $selectExistInBdd->fetch();
    if (!preg_match('/^[A-Za-z][A-Za-z0-9]{5,31}$/', $_POST['identifiant'])){
        $errors[] = "Votre nom d'utilisateur n'est pas conforme, il doit contenir entre 5 et 31 charactères sans charactères spéciaux";
    }elseif ($selectExistInBdd != NULL){
        $errors[] = "Cet identifiant est déjà utilisé";
    }
}else{
$errors[] = "Vous n'avez pas entré de nom d'utilisateur";
}
```

Nous pouvons remarqués plusieurs choses lors de cette vérification, notamment l'utilisation d'expression régulières pour empêcher l'utilisation de symboles dans le nom d'utilisateur et de définir une plage de caractères utilisables grâce à la fonction preg_match().

Nous pouvons remarquer la même utilisation pour les mots de passes :

```
if  (isset($_POST['password']) && !empty($_POST['password'])){
    $pattern = '/^(?=.*[!@#$%^+&*-])(?=.*[0-9])(?=.*[A-Z]).{8,31}$/';
    if (!preg_match($pattern, $_POST['password'])){
        $errors[] = "Votre mot de passe n'est pas conforme, il doit contenir entre 8 et 31 charactères, avec une majuscule et un caractère spécial.";
        }
    }else{
    $errors[] = "Vous n'avez pas entré de mot de passe";
}
```

Ainsi que pour l'adresse mail :

```
if (isset($_POST['email']) && !empty($_POST['email'])){
    if (!preg_match( "/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,})$/i",$_POST['email'])){
        $errors[] = "Votre mail n'est pas conforme";
    }
}else{
    $errors[] =  "Vous n'avez pas entré de mails";
}
```

Si à chaque fois le champ est bien renseigné, non vide et que le tableau d'erreurs est vide, on peut passer à l'inscription en base de données.

Le souci majeur lors de la création d'un formulaire est la sécurité lors du traitement des données de celui-ci. C'est pourquoi j'ai utilisé plusieurs méthodes afin de sécuriser les champs.

La méthode choisie pour valider le compte d'un utilisateur et l'identifier en base de données est l'utilisation d'un token qui doit être créer au préalable avec une fonction que voici :

```
function generateToken($len){
    $letter = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
    $token = "";
    $lenStr = strlen($letter);
    for ($i = 0; $i < $len; $i++){
        $random = rand(0,$lenStr-1);
        $token .= $letter[$random];
    }
    return $token;
}
```

Ce token sera utilisé pour permettre à l'utilisateur de confirmer son compte avec le token dans le lien de confirmation

Pour insérer les données récupérés, nous les passons lors de l'envoi dans la fonction htmlspecialchars() qui permet d'éviter que les données envoyés par les utilisateurs contiennent dans des balises HTML. En effet, sans cette fonction, on pourrait par exemple effectuer des injections en JavaScript avec les balises « <script></script> ».

```
htmlspecialchars($_POST['firstname'])
```

Ensuite, afin d'amener une surcouche de sécurité, nous effectuons des requêtes préparées permettant d'éviter l'injection de requêtes SQL. Cependant, ce n'est pas le seul argument en faveur de l'utilisation de ce modèle d'instruction. Une fois analysés et compilés, ils peuvent être utilisés encore et encore par le système de base de données respectif par la suite. Par conséquence, les requêtes préparés consomment beaucoup moins de ressources et sont plus rapides que les interrogations manuelles de la base de données lorsque les tâches SQL doivent être exécutées de manière répétée.

```php
$token = generateToken(30);
$passwordHash = password_hash($_POST['password'], PASSWORD_BCRYPT);
$query = $conn->prepare("INSERT INTO Compte_client(identifiant, mot_de_passe, nom_client, prenom, mail, telephone_port, telephone_fixe, adresse_1, adresse_2,
                         VALUES(:identifiant, :mdp, :nom, :prenom, :email, :telephone_port, :telephone_fixe, :adresse1, :adresse2, :ville, :cp, :pays, :coc
$query->bindValue('identifiant', htmlspecialchars($_POST['identifiant']));
$query->bindValue('mdp', $passwordHash);
$query->bindValue('nom', htmlspecialchars($_POST['name']));
$query->bindValue('prenom', htmlspecialchars($_POST['firstname']));
$query->bindValue('email', htmlspecialchars($_POST['email']));
$query->bindValue('telephone_port', htmlspecialchars($_POST['telPort']));
$query->bindValue('telephone_fixe', htmlspecialchars($_POST['tel']));
$query->bindValue('adresse1', htmlspecialchars($_POST['adresse']));
$query->bindValue('adresse2', htmlspecialchars($_POST['adresse2']));
$query->bindValue('ville', htmlspecialchars($_POST['ville']));
$query->bindValue('cp', htmlspecialchars($_POST['cp']));
$query->bindValue('pays', htmlspecialchars($_POST['pays']));
$query->bindValue('code_validation', $token);
$query->execute();
```

II- Confirmation de l'inscription

Après s'être inscrit, un mail est envoyé à l'utilisateur de cette façon en passant le token dans l'URL :

```php
$header="MIME-Version: 1.0\r\n";
$header.='From:Dendo Jitensha <contact@dendo.fr>'."\n";
$header.='Content-Type:text/html; charset="uft-8"'."\n";
$header.='Content-Transfer-Encoding: 8bit';
$to = $_POST['mail'];
$subject = "Votre inscription a Dendo Jitensh";
$message='
        <html>
          <body>
            <div align="center">
              <a href="http://dendo-jitensha/confirmation.php?token=' . $token . '">
              Confirmez votre compte !
              </a>
            </div>
          </body>
        </html>
        ';
mail($to, $subject, $message, $header);
```

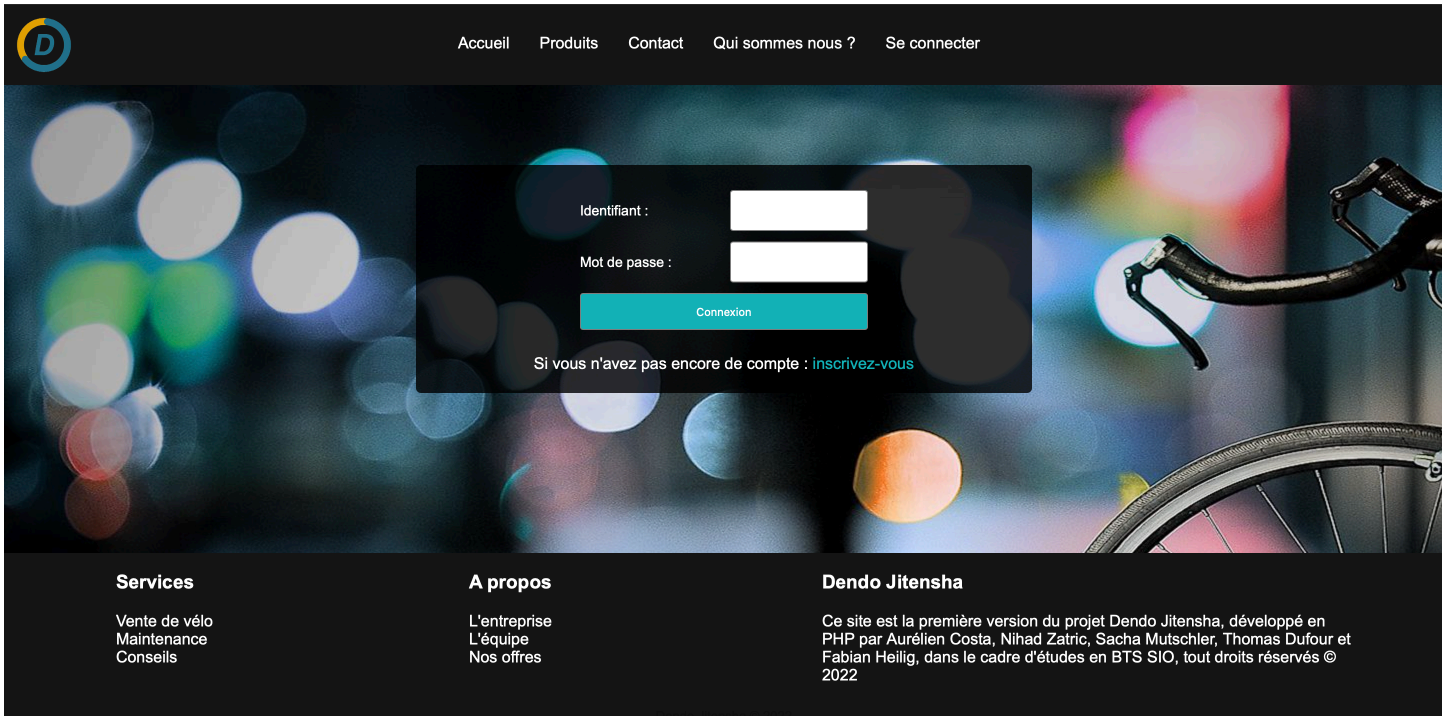
L'utilisateur recevra donc un mail avec un lien cliquable pour confirmer son compte sur la page « confirmation.php » dont voici le code :

```php
<?php
require_once ('Model/connexion_bdd.php');
$updateAccount = $conn->query('UPDATE compte_client SET etat = 1 WHERE code_validation = "' . $_GET['token'] . '" ');
echo "Votre compte a été validé";
```

Pour la confirmation de mail, le code est plutôt simple, on récupère le token passé en $_GET dans l'URL puis on le cherche dans la base de données afin de valider le compte.

III- Connexion de l'utilisateur

Après s'être inscrit et avoir validé son compte, il faut pour l'utilisateur pouvoir se connecter au site. Voici le formulaire de connexion :



Le principe d'une page de connexion est relativement simple, en effet, nous vérifions simplement les informations entrées en les comparants à ceux renseignés dans la base donnée pour définir ou non s'ils sont éligibles à la connexion.

Pour ça, il faut bien entendu, comme pour l'inscription, sécuriser les requêtes en empêchant l'utilisateur de faire de l'injection SQL et/ou Javascript dans les champs de texte.

Pour sécuriser un petit peu plus le formulaire, nous n'indiquons pas clairement si c'est le mot de passe OU l'identifiant qui est faux. Nous indiquons simplement que « le mot de passe ou l'identifiant sont incorrect ». Si nous ne faisions pas ça, un attaquant pourrait savoir si le mot de passe ou l'identifiant qu'il à rentré est correct et donc aurait une des 2 informations en sa possession.

Voici la requête afin de vérifier tout ça :

```php
if (isset($_POST['connexion']) && !empty($_POST['username']) && !empty($_POST['password'])){
    $errors = [];
    $identifiant = htmlspecialchars($_POST['username']);
    $selectUsername = $conn->prepare("SELECT * FROM compte_client WHERE identifiant = :username");
    $selectUsername->bindValue('username', $identifiant);
    $selectUsername->execute();
    $selectUsername = $selectUsername->fetch();
    if ($selectUsername == NULL){
        $errors[] = "Identifiant ou mot de passe incorrect";
    }else{
        if (!password_verify($_POST['password'], $selectUsername['mot_de_passe'])){
            $errors[] = "Identifiant ou mot de passe incorrect";
        }
    }
```

Nous avons aussi mis en place un système obligeant l'utilisateur à se connecter avant de commander un produit :

Aux cliques sur le bouton de connexion, l'ID du produit est envoyé au formulaire de connexion pour une redirection après connexion :

```php
if(isset($_POST['id_produit_connexion'])) {
    header('Location: produit.php?id_produit='.$_POST['id_produit_connexion']);
    echo'oui';
}
else {
    header('Location: index.php');
}
```

Si tout se passe bien et qu'il n'y a pas d'erreurs, on peut démarrer la session de l'utilisateur

```php
if (!$errors){
    echo "connexion reussi";
    session_start();
    $_SESSION['identifiant'] = $selectUsername['id'];
```

```php
}else{
    foreach ($errors as $error) {
        echo $error;
    }
}
```

2.5 Bilan

2.5.1 Validation des exigences point par point

Création d'un formulaire d'inscription : **FAIT**

Validation par mail : **FAIT**

Création d'un formulaire de connexion : **FAIT**

2.5.2 Axes d'amélioration

Des fonctionnalités pourraient être ajoutés notamment la réinitialisation de mot de passe par mail qui est un vrai manque sur ce projet.

Une amélioration qui pourrait être intéressant à creuser serait l'envoi de mail de confirmation par la livrairie PHP Mailer permettant d'envoyer des mails par le SMTP de Google permettant ainsi un meilleur score de mailing.

2.5.3 Compétences acquises

- Mise à niveau en requêtes SQL
- Gestion de projet avec Git
- Utilisation de GitHub
- Travail en groupe et en mode projet.