

Tables des matières

Compétences du référentiel couvertes par le projet	p 2
Résumé	p 2
Spécifications fonctionnelles	p 3
I. Arborescence du site	p 3
II. <u>Description des fonctionnalités :</u>	p 3
1. Authentification via Google	p 3
2. Authentification classique	p 3
3. Catalogue des produits et filtre	p 4
4. Fiche du produit	p 4
5. Évaluation des produits et commentaires des clients	p 4
6. Panier du client	p 4
7. Fonctionnalité de recherche	p 4
8. Espace du client	p 5
9. Suivi des commandes	p 5
10. Solution de paiement	p 5
11. Notification de paiement	p 5
12. Formulaire de contact	p 5
13. Facture du client	p 5
14. Back-office	p 6
a) Ajout et gestion des produits	p 6
b) Système de promotion	p 6
c) Suivi des commandes	p 6
d) Gestion des utilisateurs	p 7
Spécifications techniques	p 7
I. Choix techniques et environnement de travail	p 7
II. <u>Réalisations :</u>	p 8
1. Charte graphique	p 8
2. Maquette	p 8
3. Conception de la base de données	p 9
4. Extraits de code significatifs	p 10
a) Panier du client	p 10
5. Veille sur vulnérabilités de sécurité	p 13
a) Injection SQL	p 13
b) Faille XSS	p 14
c) Exposition des données sensibles	p 15
i) Comment éviter d'exposer les données sensibles en transit ?	p 15
ii) Comment éviter d'exposer les données stockées ?	p 16
6. Recherche effectuée à partir d'un site anglophone	p 17

Documents annexes	p 20
1. Maquette	p 20
2. Modèle conceptuel des données	p 21
3. Modèle relationnel des données	p 22

Compétences du référentiel couvertes par le projet

Le projet couvre les compétences énoncées ci-dessous :

- 1) Pour l'activité 1 : **Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité**
 - Maquetter une application
 - Réaliser une interface utilisateur web ou web mobile statique et adaptable
 - Développer une interface utilisateur web dynamique
 - Réaliser une interface utilisateur avec une solution de gestion de contenu ou de e-commerce
- 2) Pour l'activité 2 : **Développer la partie back-end d'une application web et web mobile en intégrant les recommandations de sécurité**
 - Créer une base de données
 - Développer les composants d'accès aux données
 - Développer la partie back-end d'une application web ou web mobile
 - Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou de e-commerce

Résumé

Il s'agit ici d'une boutique en ligne fictive pour une entreprise fictive.

L'entreprise Valse est une entreprise créée en 2022 spécialisée dans la vente de valises et de bagages et dont l'établissement se situe à Marseille. N'ayant pas de site internet, le directeur de l'entreprise m'a contacté pour réaliser une boutique en ligne correspondant à leurs attentes. L'objectif de cette boutique est d'augmenter les ventes de part une portée élargie du marché et par conséquent atteindre plus facilement de nouveaux clients.

Spécifications fonctionnelles

❖ Arborescence du site

L’arborescence du site se décline comme suit :

- Page Accueil
- Page Connexion
- Page Inscription
- Page Mon Compte
- Page Panier
- Pages Tous les produits
- Page Fiche du produit
- Page Contact
- Page Administration
- Page Informations

❖ Description des fonctionnalités

➤ Authentification via Google

L’authentification via Google est un moyen simple et rapide pour un utilisateur Google de se connecter ou créer un profil. C’est pourquoi il est intéressant de proposer une solution alternative à l’authentification classique.

➤ Authentification classique

Au-delà de l’authentification via Google, il a également été convenu de pouvoir s’inscrire et se connecter de manière classique via un formulaire d’inscription/connexion pour les clients ne souhaitant pas partager leurs informations Google.

➤ Catalogue des produits et filtre

Une page doit permettre d'afficher l'ensemble des produits disponibles. Cet affichage devra comprendre la photo du produit, le nom du produit ainsi que son prix. Un filtre doit être implémenté afin de trier les articles en fonction de leur catégorie mais également en fonction du prix, de la taille et de la couleur. Un système de pagination devrait être mis en place.

➤ Fiche du produit

L'utilisateur devra être en mesure d'accéder à une fiche du produit. Cette dernière devra comprendre :

- Une ou plusieurs photos du produit
- Le nom du produit
- Le prix
- La description du produit
- La note du produit
- Les commentaires clients

➤ Évaluation des produits et commentaires des clients

L'utilisateur devra être en mesure d'évaluer la qualité du produit et du service en général depuis un espace commentaire dédié qui se situera en dessous de la description dans la fiche du produit.

L'utilisateur devra bien entendu avoir acheté le produit en question et être connecté pour pouvoir commenter.

➤ Panier du client

L'utilisateur devra être en mesure de sélectionner un ou plusieurs articles et de mettre les articles en question dans son panier dans le but final de passer commande sur le site.

➤ Fonctionnalité de recherche

Le client devra être en mesure d'effectuer une recherche de produit ou de catégorie dans un champ prévu à cet effet. Afin de faciliter la recherche, un système de recherche avec autocomplétion doit être mis en place.

➤ Espace du client

L'utilisateur aura accès à un espace client dans lequel il lui sera possible de consulter et modifier ses informations. A savoir son nom, son prénom, son adresse email, son mot de passe (hors authentification Google) mais également son adresse postale. Il aura également la capacité de consulter ses précédentes commandes et de voir leur statut (annulée, en cours de traitement, expédiée).

➤ Solution de paiement

Il s'agit ici d'une solution de paiement fictive mais si j'avais eu à faire un projet pour une véritable entreprise j'aurais sans aucun doute choisi Stripe qui est une solution d'api bancaire simple d'utilisation.

➤ Notification du paiement

Un mail devra être envoyé au client afin de lui confirmer que sa commande a bien été passée. Ce mail devra contenir un récapitulatif de la commande ainsi qu'un lien vers la facture de la commande.

➤ Formulaire de contact

Le client devra pouvoir procéder à une réclamation si besoin à l'aide d'un formulaire de contact présent sur le site.

➤ Facture du client

Une fois la commande effectuée, le client devra pouvoir avoir accès à sa facture depuis son espace personnel avec le récapitulatif de la commande, il devra y figurer :

➤ Back-office

L'administrateur devra avoir accès à un espace sécurisé lui permettant d'administrer le site. Pour une sécurité maximale, un système de droit sera mis en place.

- Ajouter des catégories de produit

L'administrateur sera en mesure de gérer les catégories de produit c'est-à-dire créer des catégories et les supprimer.

- Gestion des produits

L'administrateur aura également la capacité de créer des produits et de les supprimer. Lors de la création du produit, ce dernier sera en mesure de :

- Donner un titre au produit
- Définir le prix du produit et même un prix promo
- Décrire le produit
- Uploader une ou plusieurs images du produit
- Établir les stocks des produits.
- Décider si ce produit doit être mis en avant ou non.

- Suivi des commandes

L'administrateur aura la possibilité de visualiser les commandes qui ont été effectuées sur le site. Ceci signifie que l'administrateur aura accès aux informations suivantes :

- La date de la commande
- Le nom et le prénom de la personne ayant passé commande
- Son adresse
- Les différents articles commandés : quantité etc.

L'administrateur pourra également changer le statut de la commande : En attente, envoyée ou annulée.

- Gestion des utilisateurs

L'administrateur devra être en mesure de supprimer un utilisateur ou de modifier le droit d'un utilisateur : possibilité de promouvoir un utilisateur au rang d'administrateur s'il le souhaite.

Spécifications techniques

❖ Choix techniques et environnement de travail

➤ Technologies utilisées pour la partie back-end :

- Le projet sera réalisé en langage PHP (Hypertext Preprocessor)
- Base de données SQL avec Power AMC

Le projet sera réalisé en Programmation Orienté Objet.

Une classe par table sera créée.

➤ Technologies utilisées pour la partie front-end :

- Le projet sera réalisé avec du HTML / CSS
- Javascript afin de dynamiser le site et améliorer l'expérience de l'utilisateur

L'environnement de développement est le suivant :

- Éditeur de code : Visual Studio Code
- Outil de versionning : Git for Desktop et Git
- Maquettage : Figma

Afin de m'organiser au mieux, j'utilisera Trello afin de découper le projet en une multitude de tâches à réaliser et de définir leur ordre de priorité.

❖ Réalisations

➤ Charte graphique

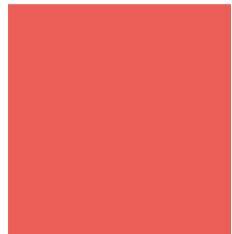
Les polices d'écriture seront les suivantes : Poppins pour le texte et Tropikal pour le logo typographique.

La couleur dominante est la suivante : #eb5e55

Charte graphique :

Poppins : Texte

Tropikal : Logo Typographique

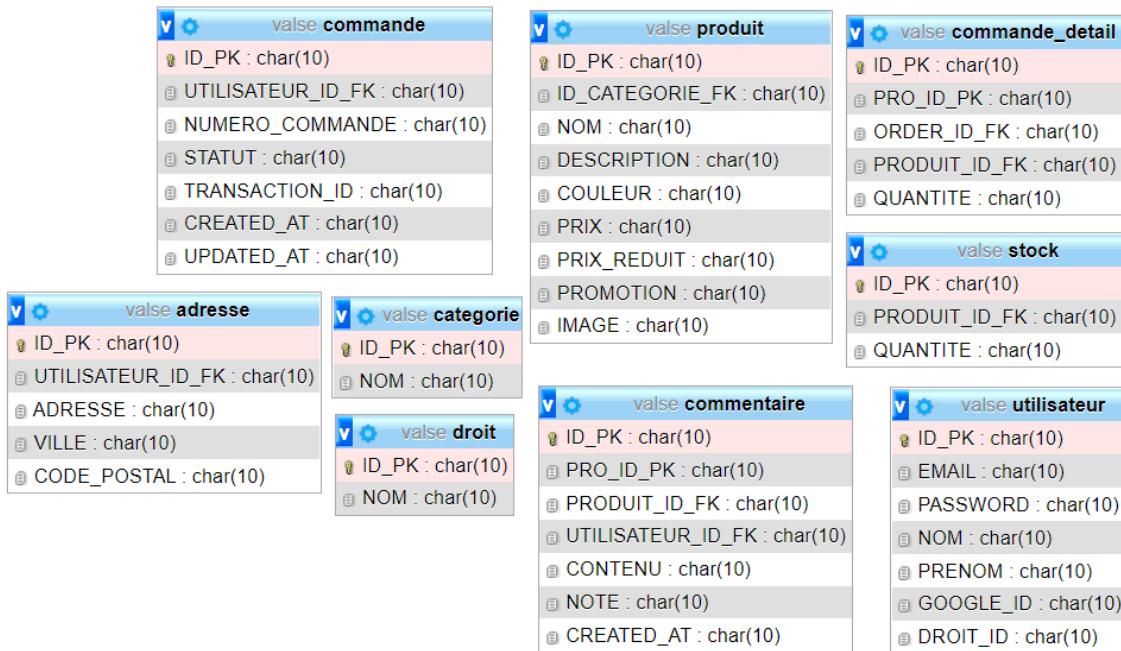


➤ Maquette

La maquette a été réalisée avec Figma. N'ayant pas de designer, l'entreprise m'a donné son aval pour concevoir l'identité visuelle ainsi que le design du site. N'étant moi-même pas designer, je me suis rendu sur Internet pour voir le design que pouvait avoir des sites web similaires afin de trouver l'inspiration nécessaire à la création de la maquette, chose que je fais souvent. Vous trouverez la maquette dans les annexes du dossier.

➤ Conception de la base de données

Au regard des fonctionnalités demandées par l'entreprise, j'ai développé la base de données suivante à l'aide de PowerAMC (Méthode MERISE) :



Vous trouverez également dans les annexes le modèle conceptuel de données ainsi que le modèle logique de données.

➤ Extraits de code significatif

▪ Panier du client

Pour la gestion du panier client j'ai choisi de stocker ce dernier en session.

J'ai donc créé un module permettant de stocker les produits dans le panier en prenant en compte la quantité entrée par l'utilisateur



```
<?php

//Module d'ajout d'articles au panier
if (!empty($_GET["action"])) {

    if (!empty($_POST["quantity"])) {

        $itemByCode = $item->getItemsbyCode($_GET["code"]);

        $itemArray = array($itemByCode[0]["code"] => array('name' => $itemByCode[0]["name"], 'code' =>
$itemByCode[0]["code"], 'quantity' => $_POST["quantity"], 'price' => $itemByCode[0]["price"], 'image'
=> $itemByCode[0]["image"]));

        if (!empty($_SESSION["cart_item"])) {

            if (in_array($itemByCode[0]["code"], array_keys($_SESSION["cart_item"]))) {

                foreach ($_SESSION["cart_item"] as $key => $value) {

                    if ($itemByCode[0]["code"] == $key) {

                        if (empty($_SESSION["cart_item"][$key]["quantity"])) {

                            $_SESSION["cart_item"][$key]["quantity"] = 0;
                        }

                        $_SESSION["cart_item"][$key]["quantity"] += $_POST["quantity"];
                    }
                }
            } else {

                $_SESSION["cart_item"] = array_merge($_SESSION["cart_item"], $itemArray);

            }
        } else {
            $_SESSION["cart_item"] = $itemArray;
            header("Location:". htmlspecialchars($_SERVER['PHP_SELF']));
        }
    }
}
```

J'affiche les données contenues dans la variable de session comme ceci :



```
<?php

$array = [];
foreach ($_SESSION["cart_item"] as $item) {

$item_price = $item["quantity"] * $item["price"];
?>

<?php $content=$item["name"] . " x " . $item['quantity'] . " : $" . $item_price . ".00";
array_push($array, $content);?>

<p><?php echo $content;?></p>
<?php

$total_quantity += $item["quantity"];
$total_price += ($item["price"] * $item["quantity"]);
}

?>
```

Puis étant donné que le panier n'est pas stocké en base de données, j'ai donc créé une classe Order contenant les fonctions permettant de :

- Récupérer les données contenues dans la variable de session et de les insérer dans la table 'commande' de la base de données une fois le paiement effectué.
- Sélectionner toutes les données d'une commande d'un utilisateur afin de les afficher directement dans son espace personnel
- Sélectionner toutes les données d'une commande d'un utilisateur afin de les afficher directement dans sa facture

```

<?php

class Order extends Dbh
{
    public function insertOrder($content, $totalPrice, $totalQuantity, $date, $code, $user)
    {
        $sth = $this->DbHandler()->prepare("INSERT INTO orders(content, totalPrice, totalQuantity,
date, code, id_user) VALUES(:content, :totalPrice, :totalQuantity, :date, :code, :id_user)");
        $sth->execute([
            ':content' => $content,
            ':totalPrice' => $totalPrice,
            ':totalQuantity' => $totalQuantity,
            ':date' => $date,
            ':code' => $code,
            ':id_user' => $user
        ]);
    }

    public function getOrder($id_user)
    {
        $sth = $this->DbHandler()->prepare("SELECT * FROM orders WHERE id_user = :id_user; ASC by id");
        $sth->execute([':id_user' => $id_user]);
        $req = $sth->fetchAll(PDO::FETCH_ASSOC);
        return $req;
    }

    public function displayOrder($code)
    {
        $sth = $this->DbHandler()->prepare("SELECT * FROM orders INNER JOIN utilisateurs ON
orders.id_user = utilisateurs.id WHERE code = :code;");
        $sth->execute([':code' => $code]);
        $req = $sth->fetchAll(PDO::FETCH_ASSOC);
        return $req;
    }
}
?>

```

```

if(isset($_POST) AND !empty($_POST) ) {
    if (!empty($_POST['code']) && !empty($_POST['crypt'])){

        if (is_numeric($_POST['code']) && is_numeric($_POST['crypt'])) {
            if (strlen($_POST['code']) <= 16 && strlen($_POST['code']) >= 16) {
                $date = date("n");
                if ($_POST['month'] >= $date) {
                    if (strlen($_POST['crypt']) <= 3 && strlen($_POST['crypt']) >= 3) {

                        $code=htmlspecialchars($_POST['code']);
                        $crypt=htmlspecialchars($_POST['crypt']);
                        $rand = rand();
                        $time = date("Y-m-d");
                        $content = implode(PHP_EOL,$array);
                        $insert->insertOrder($content, $total_price, $total_quantity, $time, $rand,
$_SESSION['current_session'][['user']]['id']);
                        unset($_SESSION['cart_item']);
                        $_SESSION['crypt'] = 123;
                    }
                    header('Location:validation.php');
                }
            }
        }
    }
}

```

Grâce à la fonction `unset()`, je détruis la variable de session ce qui a pour effet de vider le panier.

➤ Veille sur les vulnérabilités de sécurité

- Injection SQL

Qu'est-ce qu'une injection SQL ?

L'injection SQL, ou faille SQLi, est une cyberattaque réputée. Sa capacité à exploiter des failles de sécurité peut être très puissante si elle est bien exploitée.

Les attaques par injection de commandes SQL exploitent les failles de sécurité d'une application qui interagit avec des bases de données. L'attaque SQL consiste à modifier une requête SQL en cours par l'injection d'un morceau de requête non prévu, souvent par le biais d'un formulaire.

Comment s'en prémunir ?

Un moyen consiste à limiter ce que l'utilisateur peut mettre dans la zone de texte. Cela n'empêchera pas l'injection, mais c'est une mesure que vous pouvez mettre en place pour limiter des attaques de base. En effet, les caractères spéciaux spécifiques à certains langages ne pourront pas être utilisés.

A l'aide de PHP, il est possible d'utiliser certaines fonctions qui protègent les caractères spéciaux d'une chaîne de caractères pour que cette chaîne puisse être utilisé dans une requête.

Voici deux exemples :

- `addslashes()` : La fonction **addslashes()** renvoie une chaîne avec des slash devant des caractères prédéfinis.
- `real_escape_string()` : La fonction `mysql_real_escape_string()` sert à protéger les variables avant de les envoyer en base de données.

Une autre manière de se protéger contre ces attaques est de préparer les requêtes avec PDO. Ce sont des requêtes dans lesquelles les paramètres sont interprétés indépendamment de la requête elle-même. De cette manière, il est impossible d'effectuer des injections.

- Faille XSS

La faille XSS, à l'origine CSS (Cross Site Scripting) changé pour ne pas confondre avec le CSS des feuilles de style (Cascading Style Sheet), est un type de faille de sécurité des sites Web, que l'on trouve dans les applications Web mal sécurisé.

Cette faille dangereuse permet d'exécuter des scripts du côté client. Ceci signifie que vous ne pouvez exécuter que du JAVASCRIPT, HTML et d'autres langages qui ne vont s'exécuter que chez celui qui lance le script et pas sur le serveur directement. Je laisse votre imagination vous donner des idées.

Comment se protéger face à cette attaque ?

Cette faille peut-être facilement contourner à l'aide de fonctions PHP tel que htmlspecialchars() convertit les caractères spéciaux en entités HTML.*



```
require_once('classes/User.php');

//Reçoit un array contenant les info utilisateur de l'inscription pour tenter de créer un nouvel utilisateur.

function signup(array $data) : array
{
    $data = filter_input_array(INPUT_POST, FILTER_SANITIZE_FULL_SPECIAL_CHARS);

    // Filtration des données récupérées
    $login = strip_tags(strip_tags($data['login']));
    $first_name = strip_tags(strip_tags($data['first_name']));
    $last_name = strip_tags(strip_tags($data['last_name']));
    $email = strip_tags(strip_tags($data['email']));
    $password = htmlspecialchars($data['password']);
    $password_rpt = htmlspecialchars($data['password_rpt']);
```

- Exposition aux données sensibles

L'ensemble du trafic de données sur le World Wide Web a été déployé de manière ouverte, c'est-à-dire en clair et facile à pirater. C'est le protocole HTTP qui intercède pour la communication entre le navigateur et le serveur Web, sans chiffrement. Ceci facilite les activités criminelles et l'espionnage de métadonnées ou encore l'attaque Man in the Middle.

Comment éviter d'exposer les données sensibles en transit ?

Pour se prémunir contre ce problème, il faut :

- Utiliser le protocole HTTPS qui chiffrent les données sur l'ensemble des pages web
- Utilisez les requêtes GET pour récupérer les informations et POST pour modifier les informations.
- Sécuriser les cookies pour qu'ils soient transmis via HTTPS
- Sécuriser les sessions en ajoutant un délai d'expiration

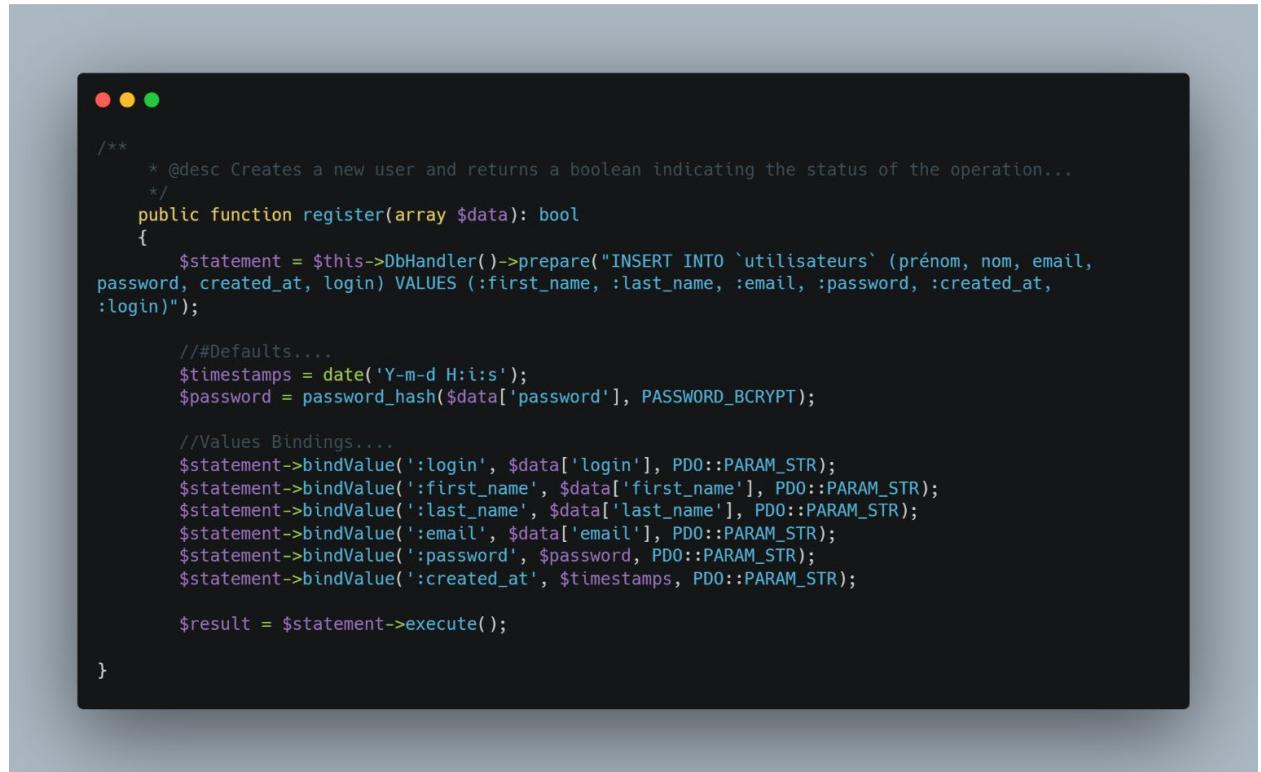
Les données sensibles ne sont pas seulement en transit, elles sont aussi stockées en base de données. Pour protéger certaines données stockées sur une application, il est possible d'utiliser des algorithmes de hachage.

L'intérêt des algorithmes de hachage est qu'ils permettent de calculer une empreint (ou hash) d'une chaîne de caractères, par exemple. Cette empreinte est utile pour éviter de stocker en clair le mot de passe dans la base de données.

Comment éviter d'exposer les données stockées ?

- Sécuriser votre base de données avec le chiffrement.
- Utiliser des algorithmes de hachage sécurisés tels que BCRYPT par exemple

Tel que dans l'exemple suivant :



```
/* @desc Creates a new user and returns a boolean indicating the status of the operation... */
public function register(array $data): bool
{
    $statement = $this->DbHandler()->prepare("INSERT INTO `utilisateurs` (prénom, nom, email,
password, created_at, login) VALUES (:first_name, :last_name, :email, :password, :created_at,
:login)");

    //Defaults....
    $timestamps = date('Y-m-d H:i:s');
    $password = password_hash($data['password'], PASSWORD_BCRYPT);

    //Values Bindings...
    $statement->bindValue(':login', $data['login'], PDO::PARAM_STR);
    $statement->bindValue(':first_name', $data['first_name'], PDO::PARAM_STR);
    $statement->bindValue(':last_name', $data['last_name'], PDO::PARAM_STR);
    $statement->bindValue(':email', $data['email'], PDO::PARAM_STR);
    $statement->bindValue(':password', $password, PDO::PARAM_STR);
    $statement->bindValue(':created_at', $timestamps, PDO::PARAM_STR);

    $result = $statement->execute();
}
```

❖ Recherche effectuée sur un site anglophone

Pour la partie contact du projet, j'ai utilisé PHPMailer qui est une librairie PHP permettant d'envoyer des mails en passant par un serveur SMTP.

Mais pour pouvoir correctement prendre en main cet outil, j'ai dû me documenter avant.

Pour se faire, je me suis rendu à l'adresse suivante :

<https://www.geeksforgeeks.org/how-to-send-an-email-using-phpmailer/>

Ils expliquent très clairement en anglais comment configurer PHPMailer et paramétrer le serveur SMTP.

Using PHPMailer:

Import the PHPMailer classe into the global namespace.

Note: Make sure that these lines are at the top of the script not inside any function.

```
use PHPMailer\PHPMailer\PHPMailer;  
use PHPMailer\PHPMailer\Exception;
```

Load the composer's autoloader.

```
require 'vendor/autoload.php';
```

Create a PHPMailer class object.

```
$mail = PHPMailer()
```

Configure the server settings:

- **SMTPDebug:** Used to display messages regarding problems in connectivity and sending emails. It has following values:
 - 0: It is default value. Disable debugging.
 - 1: Display output messages sent by the client.
 - 2: As 1, plus display responses received from the server.
 - 3: As 2, plus more information about the initial connection – this level can help diagnose STARTTLS failures.
 - 4: As 3, plus display even lower-level information.
- **isSMTP():** Set mailer to use SMTP.
- **isMail():** Set mailer to use PHP's mail function.
- **Host:** Specifies the servers.
- **SMTPAuth:** Enable/Disable SMTP Authentication.
- **Username:** Specify the username.
- **Password:** Specify the password.
- **SMTPSecure:** Specify encryption technique. Accepted values 'tls' or 'ssl'.
- **Port:** Specify the TCP port which is to be connected.

Je suis donc parvenu à l'installer correctement.

Voici d'ailleurs mon formulaire de contact :



```
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'PHPMailer/src/Exception.php';
require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';

if(isset($_POST)) {

    $masque = "/^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+\.[a-zA-Z]{2,4}$/";
    if(!preg_match($masque, $_POST['email'])) {
        echo '<p class="error">' . "L'adresse email est invalide." . '</p>';
    }
    else {
        $mail = new PHPMailer();
        $mail->SMTPDebug = 0;
        $mail->isSMTP();
        $mail->Host = "smtp.gmail.com";
        $mail->SMTPAuth = true;
        $mail->Username = "valsecontact000@gmail.com";
        $mail->Password = "laplateforme.io";
        $mail->Port = 587;
        $mail->From = $_POST['email'];
        $mail->FromName = $_POST['name'] . ' ' . $_POST['lname'];
        $mail->smtpConnect(
            array(
                "ssl" => array(
                    "verify_peer" => false,
                    "verify_peer_name" => false,
                    "allow_self_signed" => true
                )
            )
        );
        $mail->addAddress("valsecontact000@gmail.com", "Valse Contact");
        $mail->isHTML(true);
        $s = htmlspecialchars_decode('nbsp;');
        $mail->Subject = "Nouveau message de " . $_POST['name'];
        $mail->Body = $_POST['object'] . "<br><br>Message de : " . $_POST['name'] . ' ' .
        $_POST['lname'] . '<br><br>Pour contacter cette personne : ' . $_POST['email'] . '<br><br>' .
        $_POST['message'];
        $mail->send();
        echo '<p class="success">Message envoyé avec succès</strong></p>';
    }
}
?>
```

Mais comme vous pouvez le constater j'ai essayé de passer par le serveur SMTP de Google.

Ça n'a malheureusement pas fonctionné, ce qui par conséquent m'a poussé à chercher mes réponses sur internet en cherchant les mots clés suivants : smtp google phpmailer problem connection

Je suis alors tombé sur cet article :

3 Reasons why Gmail SMTP is not working with PHPMailer()

#gmail #smtp #phpmailer #email

If you are using Gmail as an SMTP Server, most chances are it won't work on the first try. And it would definitely won't work if you have enabled 2-factor authentication.

So here are the 3 steps you would need to complete before be able to send emails using Gmail SMTP Server.

1. Enable IMAP

This is simple, Just go to Gmail Settings and click the Enable IMAP option.



2. Allow Less Secure Apps

Now, go to the following link to Allow Less Secure Apps to Access your Gmail Account. <https://myaccount.google.com/lesssecureapps>

3. Unlock Other Apps

If you are using 2-Step Verification, the above step won't work. Instead you would need to visit the following link and Unlock Other Apps:

<https://accounts.google.com/DisplayUnlockCaptcha>

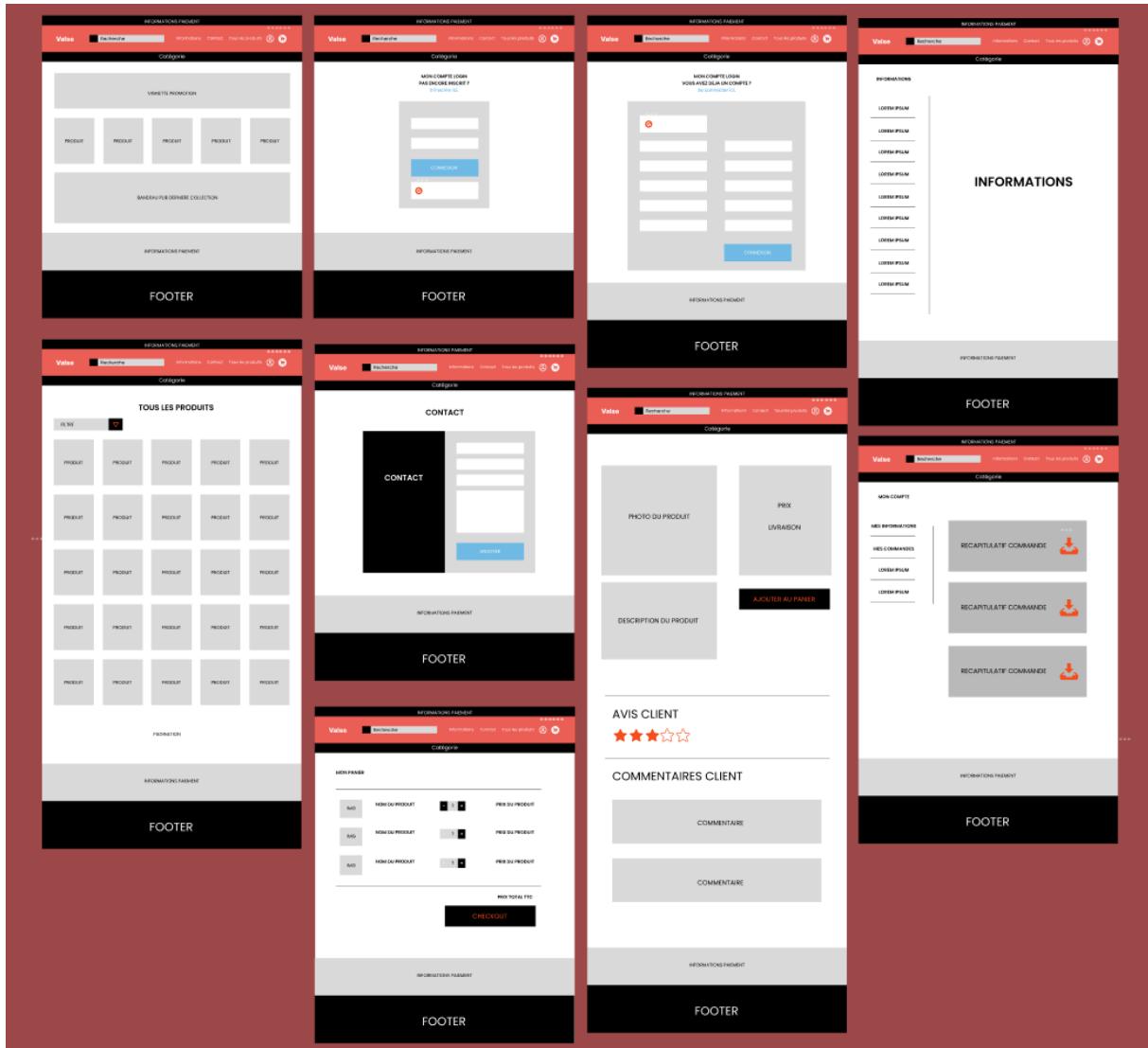
Source : <https://dev.to/surjithctly/3-reasons-why-gmail-smtp-is-not-working-with-phpmailer-d8e>

Après avoir compris que le problème venait d'un paramètre de sécurité à désactiver qu'a mis en place Google. J'ai pu tout simplement résoudre cette erreur et cela fonctionnait parfaitement.

Annexes

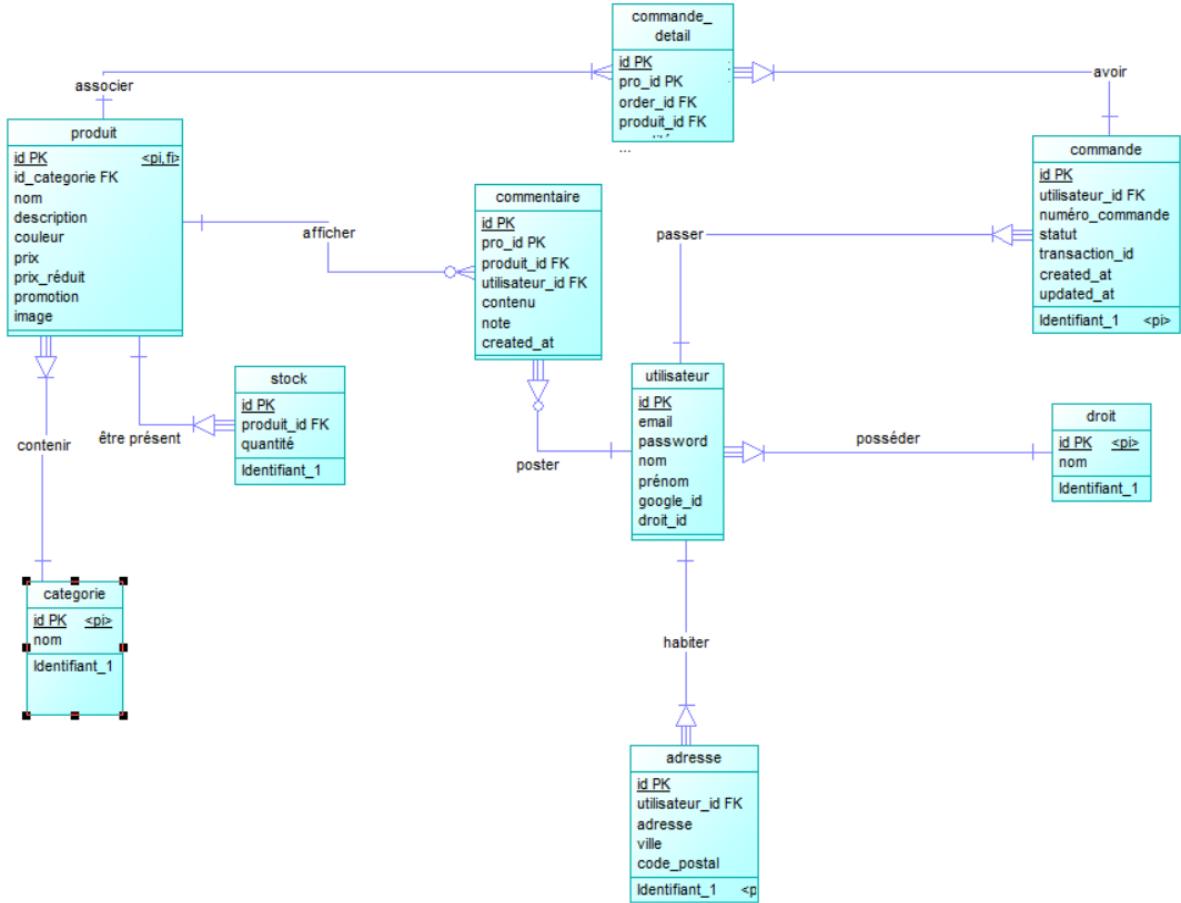
❖ Maquette

Voici la maquette du projet :



❖ Modèle logique de données

Voici le modèle logique de données :



❖ Modèle conceptuel de données

Voici le modèle conceptuel de données :

